A Guide to Automation Services 8.1

Contents

Legal notices

Introduction

This Guide explains the functionality of Automation Services for Quark Publishing System®
(QPS®) 8.1x.

Where we're coming from

This book assumes you are familiar with your computer and know how to:

• Launch an application

• Open, save, and close files

• Use menus, dialog boxes, and palettes

• Use the mouse, keyboard commands, and modifier keys

If you need help performing any of these tasks, consult the documentation resources (user
or reference guides) provided with your computer.

Conventions in this book

Formatting conventions in this guide highlight information to help you quickly find what
you need.

• Bold type style: The names of all dialog boxes, fields, and other controls are set in bold
type. For example: "Click **Storage** in the **Administration** pane."

• References: In descriptions of features, parenthetical references guide you in accessing
those features. For example: "The **System Storage** controls (**Administration** pane) let you
designate asset storage."

• Arrows: You will often see arrows (>), which map the path to a feature. For example:
"Choose **Administration** > **User Profiles** to add a user."

• Icons: Although many tools and buttons are referenced by name, which you can see by
displaying ToolTips, in some cases icons are shown for easy identification.

- Cross-platform issues: Some labels, buttons, key combinations, and other aspects of QPS client applications differ between Mac OS® and Windows® because of user interface conventions or other factors. In such cases, both the Mac OS and Windows versions are presented, separated by a slash, with the Mac OS version presented first. For example, if the Mac OS version of a button is labeled **Select**, and the Windows version is labeled **Browse**, you are directed to "Click **Select/Browse**." More complex cross-platform differences are mentioned in notes or parenthetical statements.

Notes provide helpful information about particular features and general techniques for using the software.

Understanding Automation Services

The topics below provide a high-level introduction to Automation Services.

What is Automation Services?

In a nutshell, Automation Services is a QPS service that automatically runs pre-defined workflows on a regular schedule or in response to QPS events or Web services requests. The purpose of Automation Services is to allow System Integrators to create versatile automated solutions for customers without scripting.

Like scripting, Automation Services lets you automate highly complex tasks. Unlike scripting, however, Automation Services lets you do so with a simple visual interface. Instead of having to pass a problem off to a developer so that he or she can write the necessary scripts, a System Integrator can implement and maintain solutions which automate common user scenarios himself or herself.

➡ Automation Services is versatile enough to automate most publishing tasks, but there are some things you can do with scripting that you cannot do with Automation Services. For such procedures, you can use the Automation Services "Execute QPS Script" workflow.

Here are some of the key concepts for Automation Services.

- **Workflows:** A workflow defines a sequence of actions that Automation Services can perform, such as creating a QuarkXPress project based on an XML input using a template. Workflows are generic in that they do not specify which files to work with, where to put the output, or when to execute. Automation Services ships with a set of workflows that address some of the most frequently requested automation use cases that QPS customers implement. For unique situations, System Integrators can create workflows in Visual Studio using the Automation Services SDK.

- **Transformations:** A transformation is a set of instructions for converting data from one format to another. For example, a transformation might define how content in a QuarkXPress® layout should be converted to HTML for fowarding to Drupal. Transformations support XML, Excel spreadsheets, databases, QuarkXPress projects, and QuarkCopyDesk® articles as valid formats. You create a transformation in QPS Connect Client, and then Automation Services compiles the transformation into executable code for use by the Automation Services server component and deploys it to the server.

➡ In transformations, QuarkXPress and QuarkCopyDesk files are represented by XML mapping components. The QuarkXPress Server Modifier XML schema allows QuarkXPress projects and QuarkCopyDesk articles to be represented in an XML format. Automation Services for QPS uses QuarkXPress Server to convert between XML page representations and native QuarkXPress and QuarkCopyDesk file formats.

- **Workflow variables:** Variables define how a workflow will execute. For example, the variables required to execute a particular workflow might include instructions on when to execute, an input file, a transformation to execute, a location to search for image files used in the transformation, an output file name, and instructions for where to store the final file.

- **Trigger:** A trigger is an event that causes an automation profile to execute. Automation profiles can be executed manually, via Web services, on a regular schedule, or in response to a QPS event (such as a status change) occurring for a file that meets a specific set of search criteria.

- **Automation profiles:** An automation profile is where it all comes together. An automation profile says, "Execute this workflow using this transformation and these variables."

Automation Services has two components:

- The server component is a Web application executes automation profiles in response to triggers. It also provides an execution environment for the transformations used in those automation profiles. The server component can run on a separate computer from QPS Server, so that it does not meaningfully impact the performance of QPS.

- The client component is a plug-in for QPS Connect Client that adds two panes to the main window: The **Transformation** pane, which lets you create and test transformations, and the **Automation** pane, which lets you create and deploy automation profiles.

➡ Automation Services is available for Windows only.

Process overview

The process of using Automation Services is, in general:

1 Define in detail the process you want to automate.

2 Consult the Automation Services *Workflow Reference* to determine if there is an existing workflow that you can use. (If no such workflow exists, Quark or a System Integrator can use the Automation Services SDK to create one.)

3 For steps in the workflow where content needs to be converted into another format or combined with other content, create and deploy transformations in QPS Connect Client.

4 Pull everything together by creating an automation profile in QPS Connect Client.

5 Test and deploy the automation profile.

The following topics explain these steps and concepts in detail.

Understanding workflows

A workflow is a predefined sequence of actions for creating output. You can also think of a workflow as a framework on which you can build an automated task. For example, here is a list of various publishing tasks you might want to automate, with the workflows you would use to automate them. (Note that not all of these workflows are included with Automation Services.)

Publishing task	Workflow(s)
Magazines, journals, newspapers, educational books	XML to QuarkCopyDesk to QPS
Datasheets	XML to QuarkXPress to QPS
High-design technical documentation such as consumer product manuals, books, reports, and retail "magalogs"	XML, database, Excel to QuarkXPress to QPS
Technical documentation	XML to PDF via QuarkXPress
Archival of print content for database repurposing	QuarkCopyDesk to XML to database
"Print first" Web publishing	QuarkCopyDesk to HTML to Drupal
"Content first" Web publishing	XML to HTML to Drupal
Personalized retail mailers and flyers	XML Import to QuarkXPress to PDF

A workflow defines source and output types, but does not define how content should be combined and/or converted from one format to another. For example, the "XML to QuarkXPress" workflow can insert XML content into a QuarkXPress template and produce a QuarkXPress project as its output. However, it does not specify where the XML file and the QuarkXPress template come from, how the XML content should be mapped to the QuarkXPress template, or where the finished project should be put.
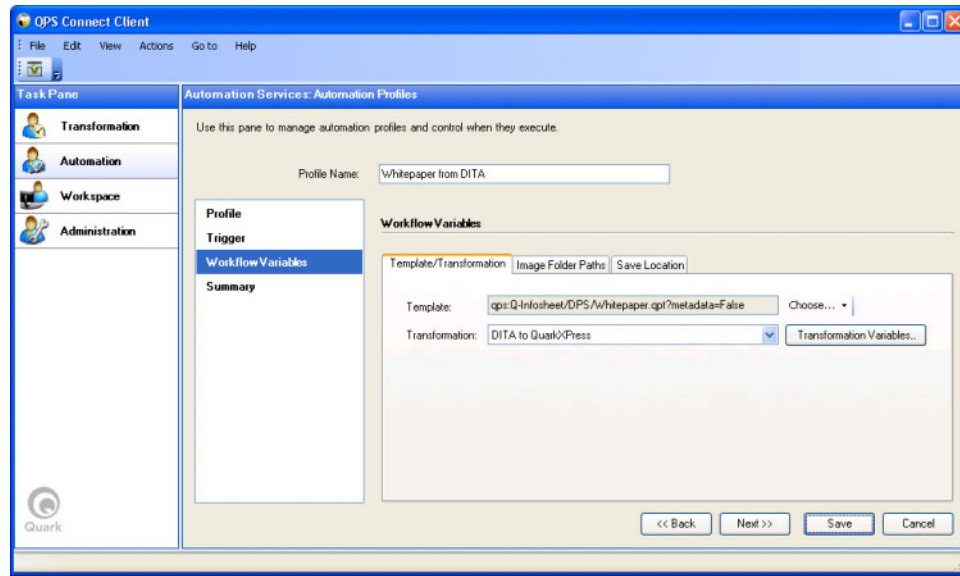
Workflow Variables screen for a workflow

The workflows provided with Automation Services include:

• XML to QuarkCopyDesk

• XML to QuarkCopyDesk From Template

• XML to QuarkXPress

• XML to XHTML (which includes the ability to transform QuarkXPress and QuarkCopyDesk content to HTML)

• XML to XML (which includes the ability to transform XML into QuarkXPress and QuarkCopyDesk content)

• Export from QuarkXPress Project (which lets you export layouts in PDF and SWF format)

• Execute QPS Script

For more information about the workflows provided by Quark, see the *Automation Services Workflow Reference*.

Understanding transformations

A transformation is the portion of an automated task that indicates exactly how content should be combined and/or converted to another format.
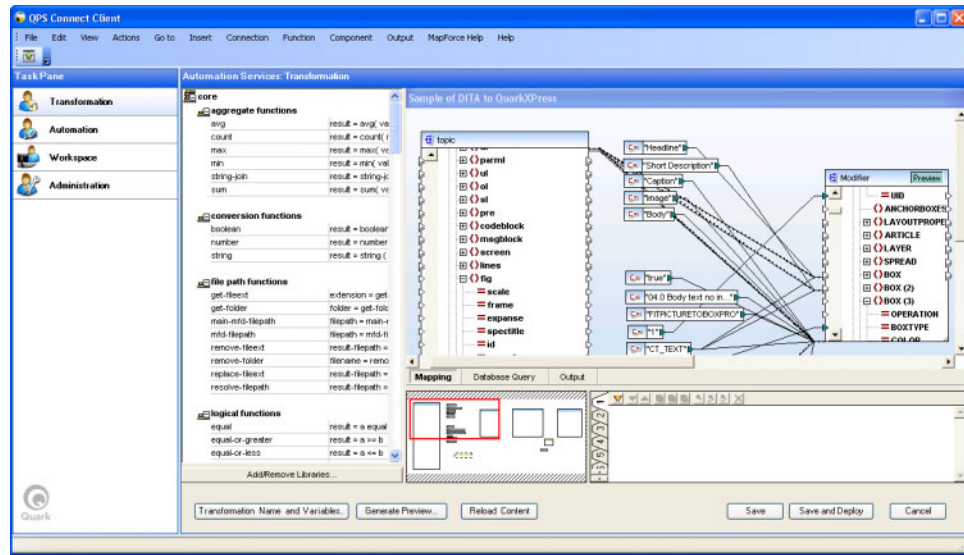
An example transformation

You can use transformations to:

- Convert XML content and data into Quark's Modifier XML format for publishing as QuarkXPress projects and QuarkCopyDesk articles.

- Insert data from various data sources into documents. Valid data formats include XML files, QuarkXPress projects and templates, QuarkCopyDesk articles and templates, and XHTML files.

- Convert XML (including Quark's Modifier XML format) to digital publishing formats, including HTML.

Valid data sources include files from the file system, QPS assets, the results of a QPS search, and files in the QuarkXPress Server document pool.

When you create a transformation, you can specify that some data sources are fixed and others are variable. Fixed variables are always used, but you can change variable inputs when you use the transformation in an automation profile.

The core of each transformation is its *map*. The map, shown in the center of the window above, lets you control which parts of the input or inputs go to which parts of the output or outputs. It also allows you to insert logic that alters the content during the transformation.

For example, assume that the inputs to a transformation profile are:

- An XML file containing brochure content. A new version of this content is created every month and checked into QPS as a new file. The schema for the file is internal to your organization.

- A spreadsheet containing customer records. This file is available over the network via a shared volume. This content of this file changes constantly, but the file name and location do not change.

- A QuarkXPress template. This file is in the document pool for a QuarkXPress Server instance. This file never changes.

To use these files, you could create a transformation that:

- Accepts an XML file that uses the internal schema. Because this file is the file that triggers the automation profile, it is referred to as the *trigger input*. Because you will be using different files at different times, you would designate this as a *variable input*.

- Maps the element types in the XML file to specific parts of the QuarkXPress template. For example, `<title>` elements in the XML might be mapped to headline boxes in the template, and `<body>` elements might be mapped to body text boxes. Because the template never changes, you could designate it as a *fixed input*.

- Maps columns in the spreadsheet to other parts of the QuarkXPress template, so that each output file can be personalized for a particular customer. For example, the "Name" column might be mapped to a box on the cover of the brochure, or the "Name" and "Address" columns might be mapped to an address box.

➡ If you specify a fixed input, and that input changes between the creation of the transformation and the time when you execute it, Automation Services uses the updated version of that input.

To create this transformation, you would need:

- The schema for the XML file that will be used.

- An optional example of the XML file that will be used. (Note that this file is used only for creating the map — this is a variable input, so even if you use an XML file here, that file will not be used when the workflow is executed.)

- Access to the spreadsheet file to be used.

- A copy of the template to be used.

While working on a transformation, you can test the output in two ways: With a preview of the transformed XML, and with a preview of the final product. Once a transformation is complete, you deploy it to the server, where it can be used in automation profiles. For more information, see "*Working with transformations*."

➡ To create a transformation's map, you use an embedded instance of Altova MapForce that runs inside of QPS Connect Client.

Construct vs. Modify

A transformation that creates a QuarkXPress project or QuarkCopyDesk article can do so in two ways:

- *Using the Construct method*. This method creates the project or article from scratch, using the default Job Jackets file for the publication and section where it will be checked in. This approach is best for situations where you don't know how many boxes or components

will need to be created. For an example of a Construct transformation, see "*XML to QuarkCopyDesk (sample transformation)*."

• *Using the Modify method*. This method creates the project or article using a template. This approach is best for situations where the number of boxes or components does not change. This approach also lets you do the bulk of the layout tasks in QuarkXPress, and use Automation Services only to fill in variable content. For an example of a Modify transformation, see "*XML to QuarkCopyDesk with Template (sample transformation)*."

Understanding automation profiles

An automation profile lets you specify when a particular workflow should be executed and control which inputs it uses. You can specify that an automation profile should execute:

• On a regular schedule.

• When a particular event occurs (for example, when an asset in QPS is routed to **PDF Output**).

• Only when you manually execute it.

• Only when triggered by a Web services call.
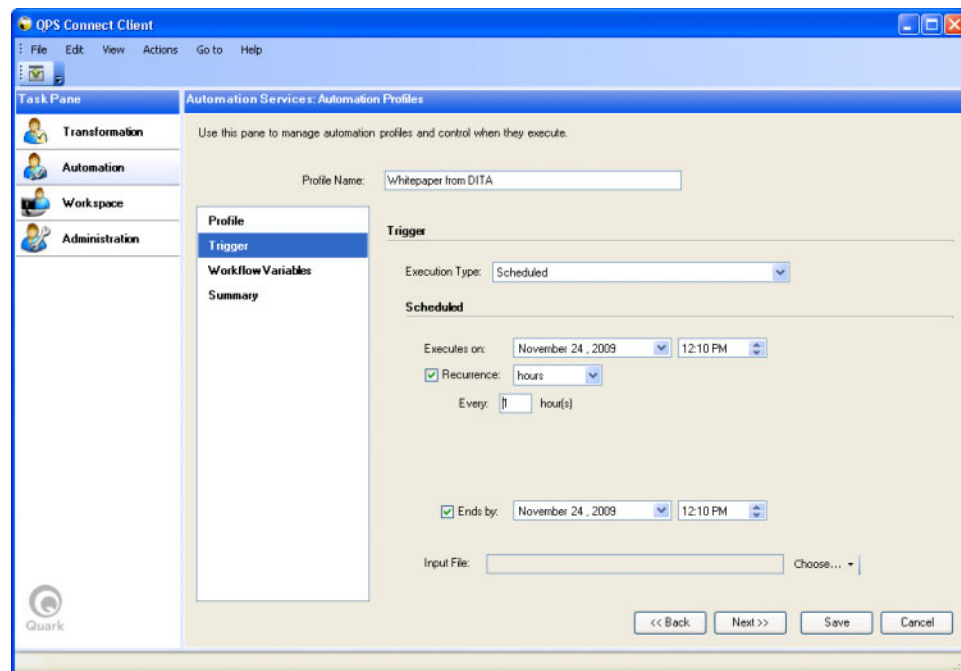
Trigger screen for an automation profile set to execute on a schedule

An automation profile also lets you configure *workflow variables*. For example, if you are using the "XML to QuarkXPress Template" workflow, the workflow variables might include:

• **Trigger Input**: For event-driven automation profiles, the trigger input is the asset that triggers the automation profile's execution. The trigger input consists of an event plus a

trigger asset filter. When one of the events you specify occurs for an asset that matches the trigger asset filter, the automation profile executes.

- **Transformation**: The transformation that should be used.

- **Template Name**: The file name for the specific template to be used (if the workflow requires a template).

- **Image Search Folders**: Folders where Automation Services can look for picture files used in a transformation.

- **Save Location**: The place where the completed file should be put. This could be a folder in the file system or a location in QPS.

Note that the above variables are provided only as examples. The specific workflow variables that you must supply differ from workflow to workflow. For more information, see "*Working with automation profiles*."

Starting and stopping Automation Services

The server component of Automation Services runs as a Web application inside an application pool in Microsoft Internet Information Services (IIS). To stop or start the server component, open the IIS management console, right-click on the application pool for Automation Services, and choose **Start**, **Stop**, or **Recycle** from the context menu.
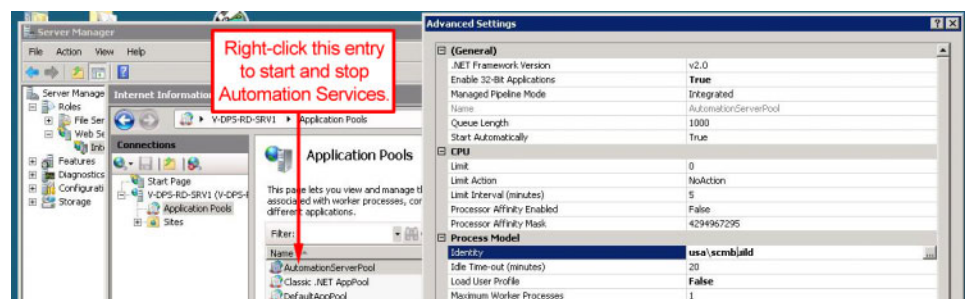
IIS management console with Automation Services application pool displayed. The **Advanced Settings** dialog box displays when you right-click the **AutomationServerPool** icon and choose **Advanced Settings**.
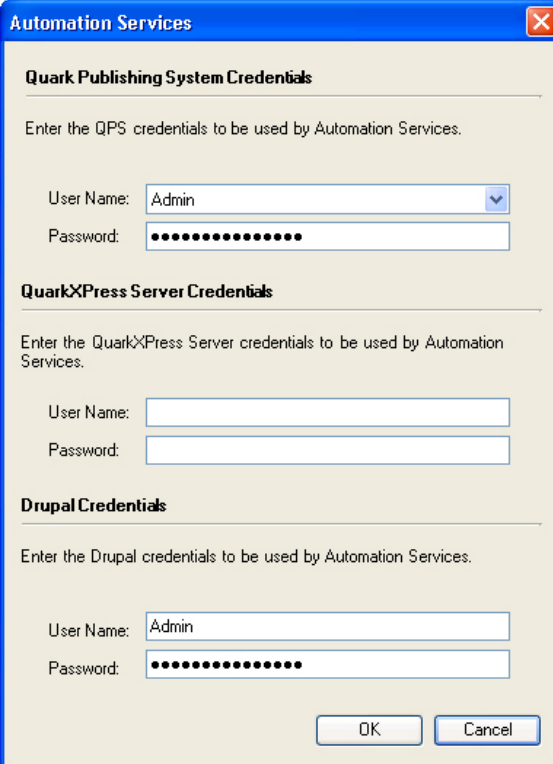
To start the server, you must also do one of the following things:

- Launch QPS Connect Client and log in.

- Go to the following URL in a Web browser, using the IP address and port number of the Automation Services server computer: `http://[IP address]:[port]/Quark Automation Services/ProfileManagementService.svc`. If this page displays correctly, the server is up and running.

Setting log on credentials

In order to interact with QPS, QuarkXPress Server, and Drupal, Automation Services must have appropriate log-on credentials for each. To set up user accounts for Automation Services:

1 In the QPS Connect Client **Automation** pane, choose **Edit > Automation Server Preferences**. The **Automation Services Preferences** dialog box displays.

Automation Server Preferences dialog box

➡ If this menu item is not available, it means no server has been detected. Verify that the server component of Automation Services is running.

2 Enter user names and passwords for QPS, QuarkXPress Server, and Drupal, then click **OK**.

The Automation Services user interface

The topics below provide a reference for the Automation Services user interface in QPS Connect Client.

Automation pane

The **Automation** pane lets you create, configure, and execute automation profiles. (For more information, see "*Working with automation profiles*.")
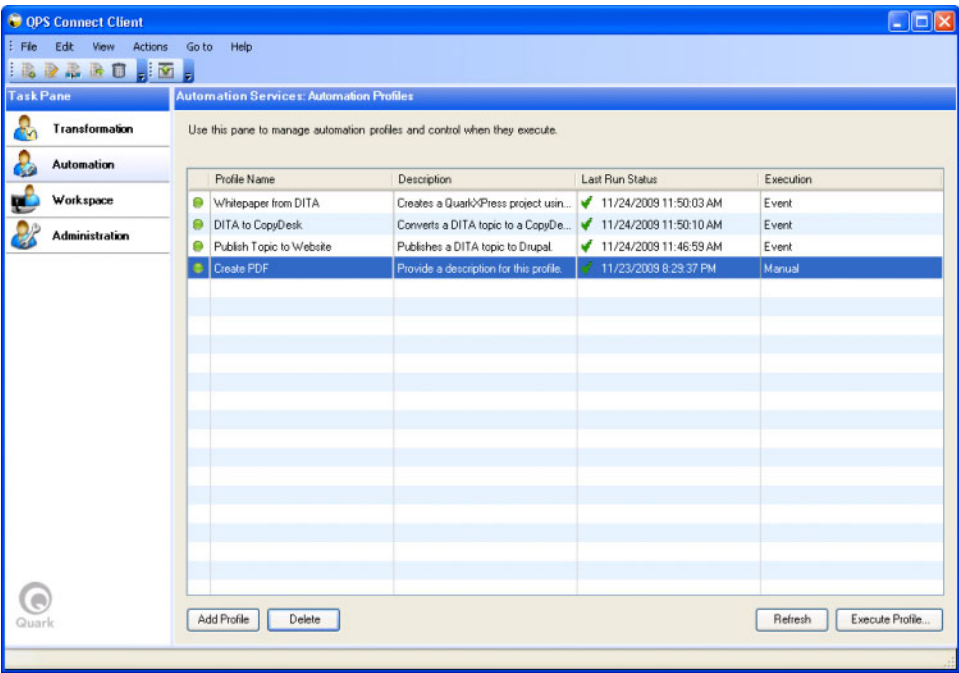
Automation pane of **QPS Connect Client** window

The **Automation** pane includes the following columns:

- Enabled column: Indicates which processes are disabled (gray) and enabled (green). To enable or disable an automation profile, right-click its icon in this column and choose **Enable** or **Disable** from the context menu.

- **Profile Name** column: Lists each automation profile by name.

- **Description**: Describes each automation profile.

- **Last Run Status** column: Indicates the date and time when each automation profile was last run. A green checkmarks indicates that no errors occured. A yellow caution icon indicates an error. To view the error message, mouse over the yellow icon.

- **Execution** column: Indicates when each automation profile will execute. The options are **Manual**, **Automatic**, and **Event** (event-triggered).

You can sort each column by clicking its label, and reverse the sort direction by clicking the label again.

The buttons at the bottom of the pane are as follows:

- **Add Profile**: Creates a new automation profile and displays the **Add Profile** screen. For more information, see "*Add Profile screen*."

- **Delete**: Deletes the selected automation profile.

- **Refresh**: Refreshes the information in the **Last Run Status** column.

- **Run**: Runs the selected automation profile.

➡ Most of the functionality described in this topic is also available by right-clicking an automation profile.

Add Profile screen

The **Add Profile** screen lets you specify a workflow and name for an automation profile.
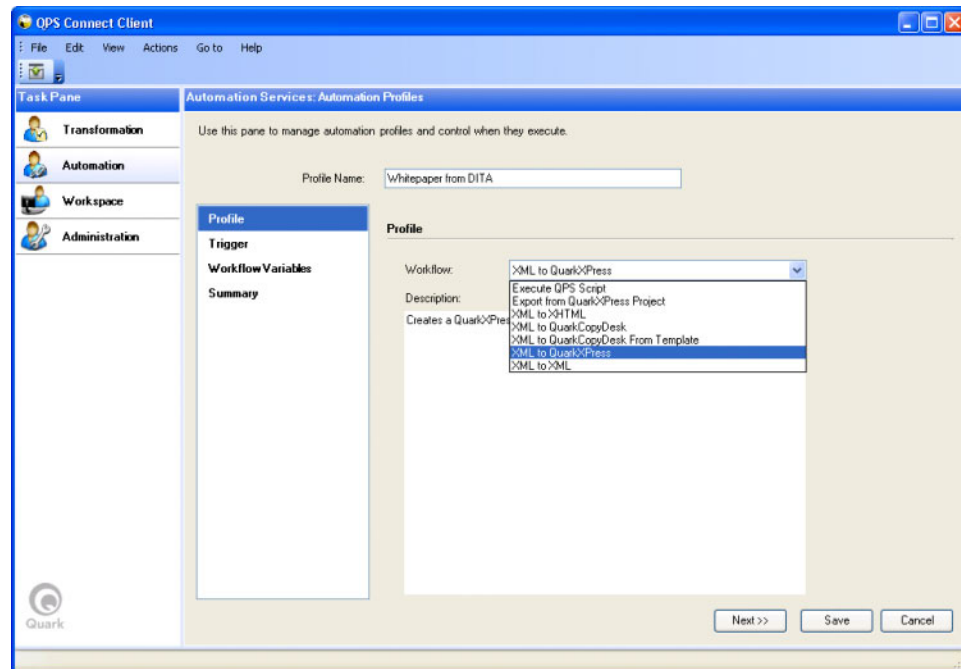
Add Profile screen of **Automation** pane

This screen contains the following controls:

- **Automation Steps** area: Lets you move to different screens within the task of creating an automation profile.

- **Select Workflow** drop-down menu: Lets you choose a workflow to use in the automation profile. For more information, see "*Understanding transformations*."

- **Profile Name**: Lets you specify a name for the automation profile.

- **Description** field: Lets you specify a description for the automation profile. The description is displayed in the main screen of the **Automation** pane.

- **Next** button: Advances to the **Add Scheduling** screen.

- **Save** button: Saves the current information without leaving the screen.

- **Cancel** button: Discards all information and returns to the main screen of the **Automation** pane.

Trigger screen

The **Trigger** screen lets you schedule when an automation profile runs.
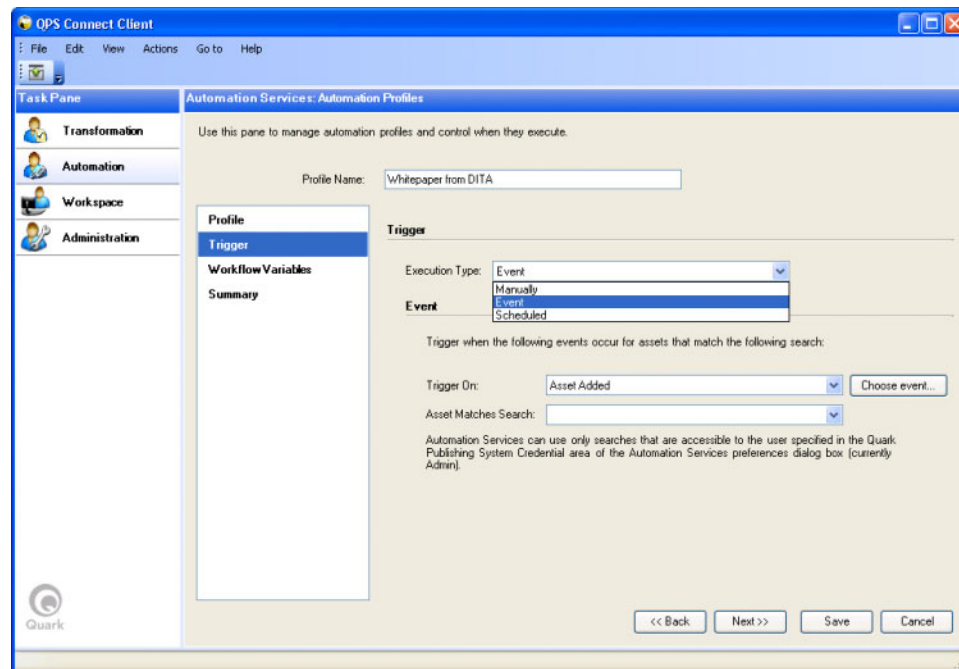
Add Scheduling screen of **Automation** pane

This screen contains the following controls:

- **Automation Steps** area: Lets you move to different screens within the task of creating an automation profile.

- **Profile Name** drop-down menu: Indicates the name of the automation profile.

- **Execution Type** drop-down menu: Lets you specify whether the automation profile is scheduled **(Scheduled)**, event-triggered **(Event)**, or manually executed **(Manually)**. The lower portion of the screen changes depending on this selection.

- **Back** button: Returns to the **Add Profile** screen.

- **Next** button: Advances to the **Add Scheduling** screen.

- **Save** button: Saves the current information without leaving the screen.

- **Cancel** button: Discards all information and returns to the main screen of the **Automation** pane.

Scheduled tasks

When **Scheduled** is selected for **Execution Type**, the screen looks like this:

Scheduled area

This area contains the following controls:

- **Executes on** controls: Lets you set a date and time for the execution of the automation profile.

- **Recurring** controls: Lets you configure the automation profile to run periodically.

- **Ends by** controls: Lets you specify an end date for recurring execution.

Event-triggered tasks

When **Scheduled** is selected for **Execution Type**, the screen looks like this:

Event area

This area contains the following controls:

- **Trigger On** drop-down menu: Lets you choose an event to trigger the automation profile. The **Choose event** button to the right displays the **Choose Events** dialog box, which lets you select multiple events.

Select Events dialog box

- **Asset Matches Search** field: Lets you choose which search the trigger asset should match. The **Manage Searches** menu item displays the **Manage Saved Searches** dialog box, which lets you create or edit a saved search.
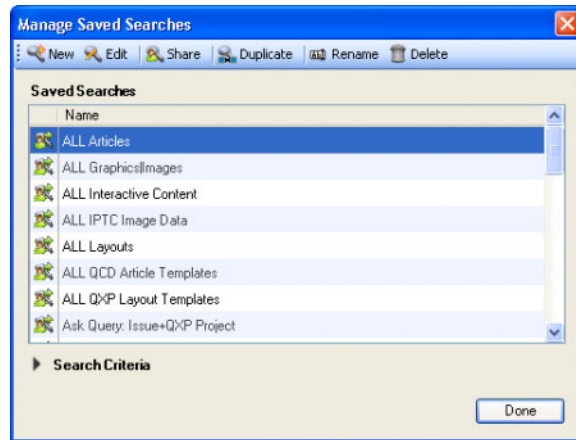
Manage Saved Searches dialog box

Manual tasks

When **Manually** is selected for **Execution Type**, the screen looks like this:
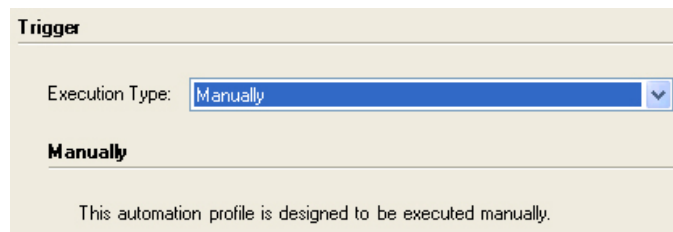
Manually area

Workflow Variables screen

The **Workflow Variables** screen lets you configure the variables used by an automation profile. The controls shown in this screen are defined by the workflow and differ from workflow to workflow. For more information, see the *Automation Services Workflow Reference*.
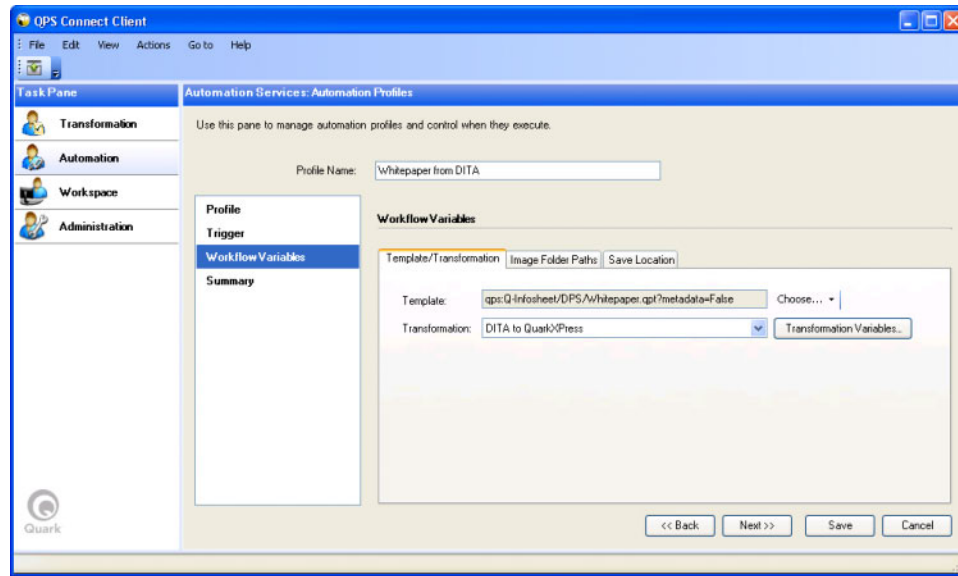
Workflow Variables screen of **Automation** pane

Summary screen

The **Summary** screen provides a summary of the automation profile.
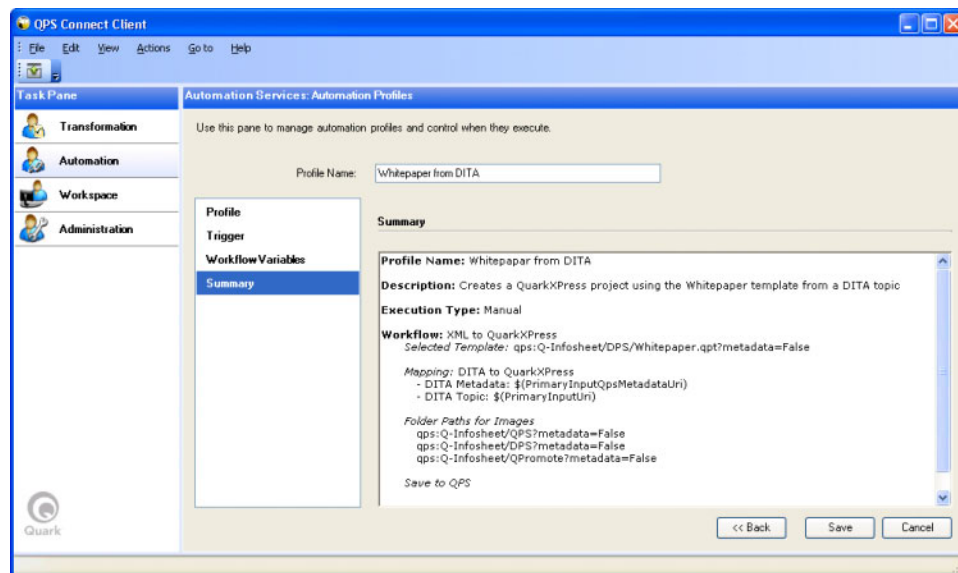
Summary screen of **Automation** pane

This screen contains the following controls:

- **Automation Steps** area: Lets you move to different screens within the task of creating an automation profile.

- **Back** button: Returns to the **Add Scheduling** screen.

- **Finish** button: Saves the automation profile and returns to the main screen of the **Automation** pane.

- **Save** button: Saves the current information without leaving the screen.

- **Cancel** button: Discards all information and returns to the main screen of the **Automation** pane.

Transformation pane

The **Transformation** pane lets you create and configure transformations. (For more information, see "*Working with transformations*.")
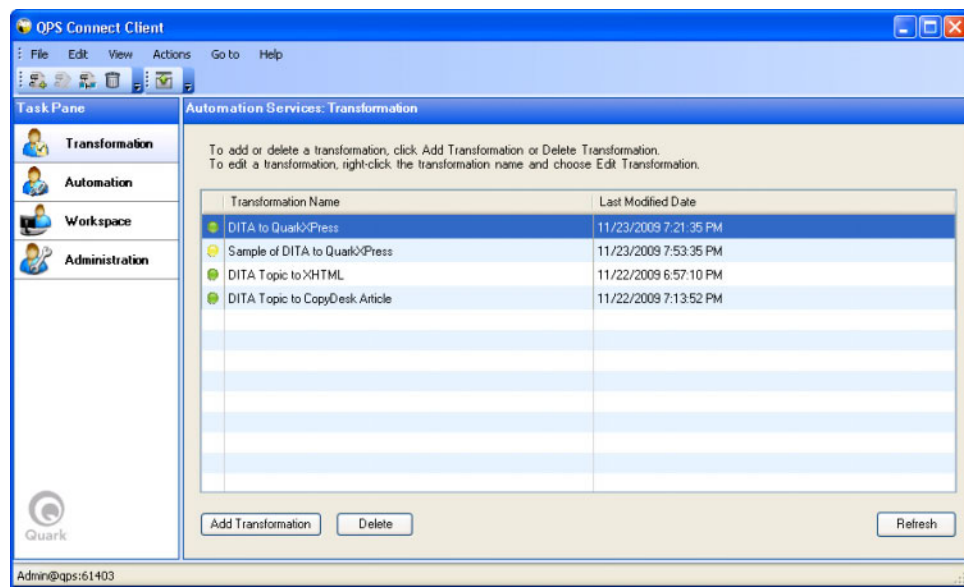
Transformation pane of **QPS Connect Client** window

The **Transformation** pane includes the following columns:

- Status column: Indicates which transformations are incomplete (gray) and deployed (green).

- **Transformation Name** column: Lists each transformation by name.

- **Last Modified Date**: Indicates when each transformation was last modified.

You can sort each column by clicking its label, and reverse the sort direction by clicking the label again.

The buttons at the bottom of the pane are as follows:

- **Add Transformation**: Creates a new transformation and displays the **Edit Transformation** screen. For more information, see "*Edit Transformation screen*."

- **Delete Transformation**: Deletes the selected transformation. Note that you cannot delete a transformation that is being used by an automation.

- **Deploy Transformation**: Compiles the selected transformation and deploys it to the server. Only deployed transformations can be used in automation profiles.

➡ Most of the functionality described in this topic is also available by right-clicking a transformation.

Edit Transformation screen

The **Edit Transformation** screen lets you configure a transformation.
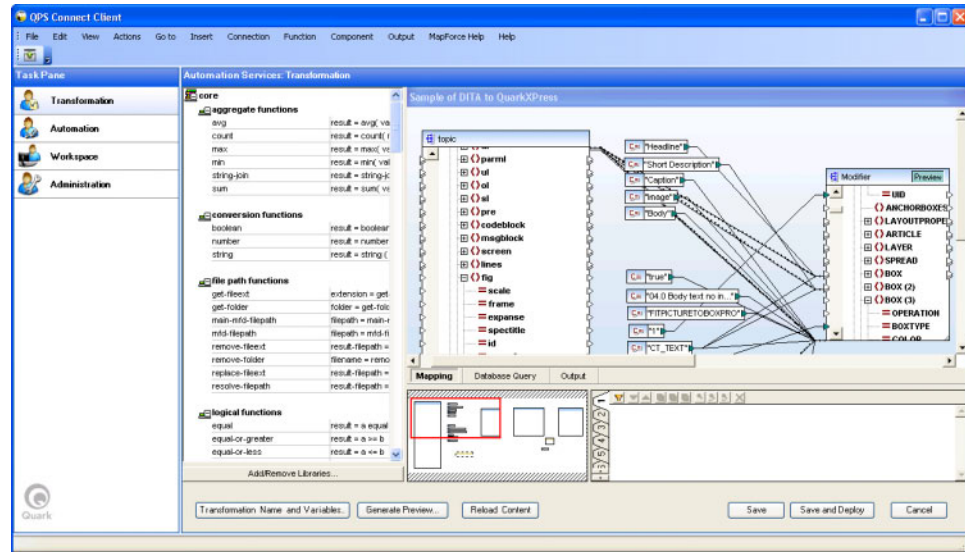
Edit Task screen

For information about the main controls in the **Edit Task** screen, see the Altova MapForce documentation.

In addition, this screen includes the following controls:

- **Generate Preview** button: Displays the **Generate Preview** dialog box, which lets you create a preview of the output of the transformation in its current state, using the inputs specified in the **Transformation Name & Variables** dialog box.
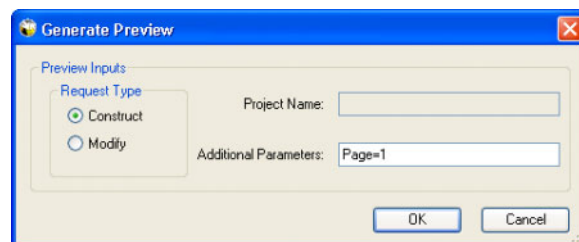
Generate Preview dialog box

- **Save**: Displays a dialog box that lets you name or rename the transformation, then saves the transformation without deploying it to the server.

- **Save and Deploy**: Displays a dialog box that lets you name or rename the transformation, then saves the transformation, compiles it, and deploys it to the server. Only deployed transformations can be used in automation profiles.

➡ You cannot modify a deployed transformation.

- **Back** button: Returns to the main screen of the **Transformation** pane.

- **Next** button: Advances to the **Deploy Task** screen.

Transformation Name and Variables dialog box

The **Transformation Name & Variables** dialog box lets you select parameters that you want to be varaible when the transformation is used.
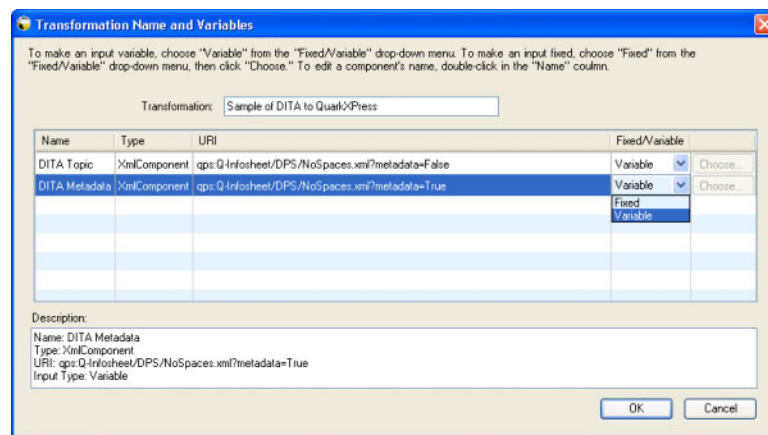
Transformation Name & Variables dialog box

Enter a name for the transformation in the **Transformation** field.

The **Name** column lets you specify a user-friendly name for each input. This user-friendly name is what displays when you are using the transformation in an automation profile.

The **Type** column indicates each input's type.

The **URI** column displays the path of each input

The **Fixed/Variable** drop-down menu lets you control which inputs are fixed (compiled into the transformation) and which ones are variable (can be specified in an automation profile).

The **Choose** button, which is available only for fixed inputs, lets you specify which specific input should be compiled into the transformation for fixed inputs.

The **Description** field shows information about whichever input is selected above.

Working with transformations

The topics below describe how to create and use transformations.

➡ The QPS Server you log on in QPS Connect Client must be the same server as the one specified in the "AutomationServices.Server.config" file (located at `[application folder]\Server\AutomationServices_Data\Resources\Configuration`). If it is not, the assets and searches you use to build an automation profile may not be available when that automation profile executes.

Creating and deploying a transformation

To create and deploy transformation:

1 In QPS Connect Client, display the **Transformation** pane.
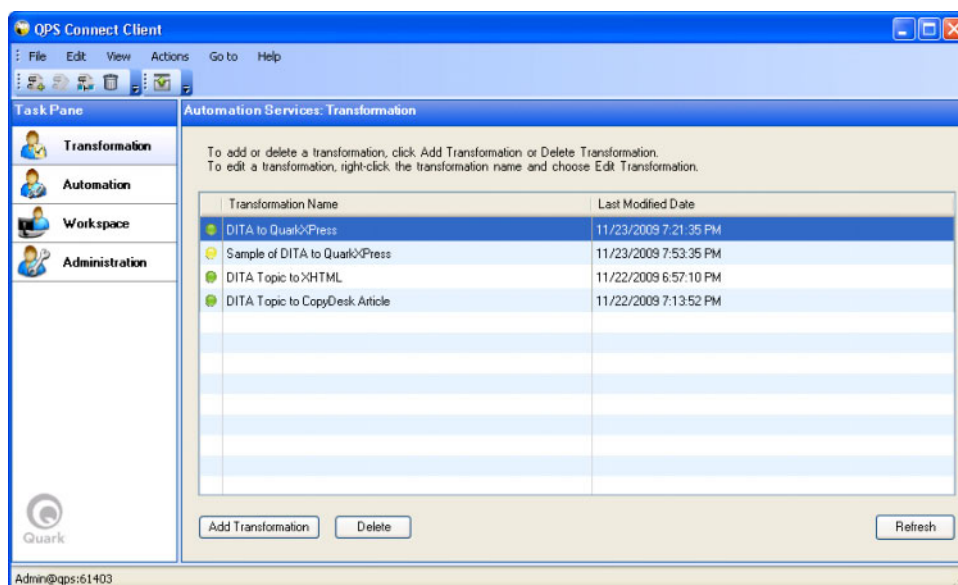
Transformation pane of **QPS Connect Client** window

2 Click **Add Transformation**. The **Edit Transformation** screen displays.
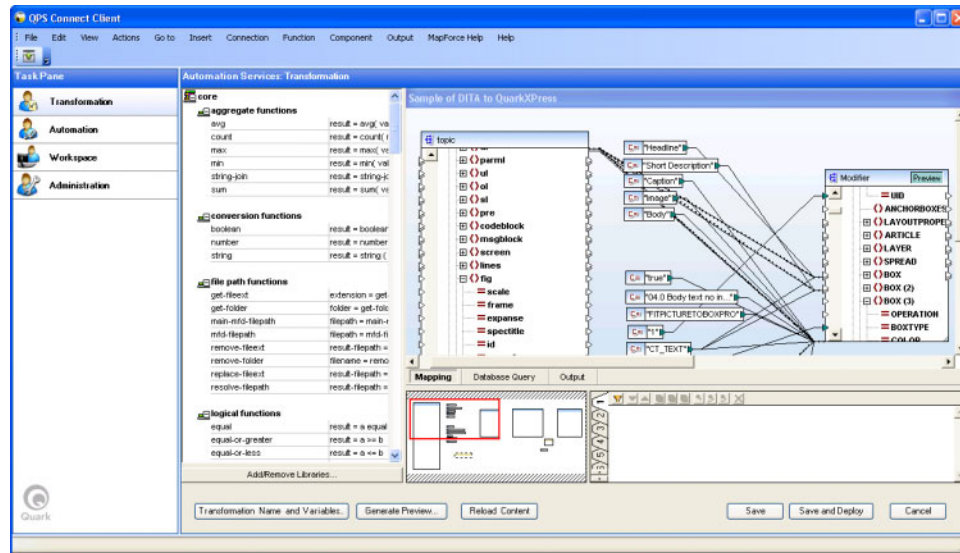
Edit Transformation screen

➡ For information about the main controls in the **Edit Transformation** screen, see the Altova MapForce documentation. For information about the other controls, see "*Transformation pane*."

3 To specify the first input for the transformation, choose **Insert > XML Component/File**. The **Insert XML Component** dialog box displays.
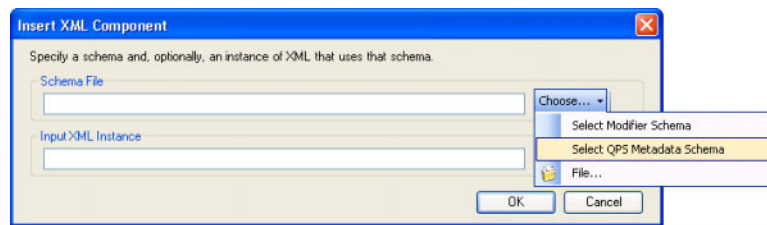
Insert XML Component dialog box

4 Use the **Choose** menu to the right of the **Schema File** field to choose **Select Modifier Schema** (if the input will be a QuarkXPress project or QuarkCopyDesk file), **Select QPS Metadata Schema** (if the input will be QPS metadata), or **File** (if the input will be an XML schema). The **Schema File** field updates automatically to show the path of the schema file.

➡ Automation Services supports both DTDs and XML Schema.

5 Use the **Choose** menu to the right of the **Input XML Instance** field to choose a file that uses the schema you selected in the **Schema File** field. You can choose a file in the file system, in QPS, or in the QuarkXPress Server document pool.
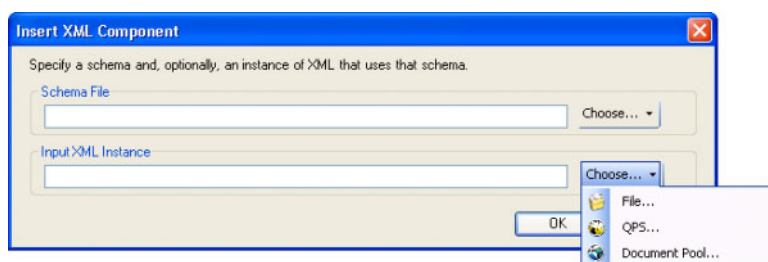
Insert XML Component dialog box

➡ Make sure that when Automation Services runs, it has access to the input you specify here. If it does not, it will use the input you specified when you created the transformation.

➡ If a fixed input or variable input is not available when a transformation executes, the transformation fails.

6   If you choose **QPS**, the **QPS Asset Selection** dialog box displays. By default, this dialog box shows all projects and articles in all sections. To search for a particular asset, drag down from the top of the **Workspace: Search Results** area to expose the search controls.
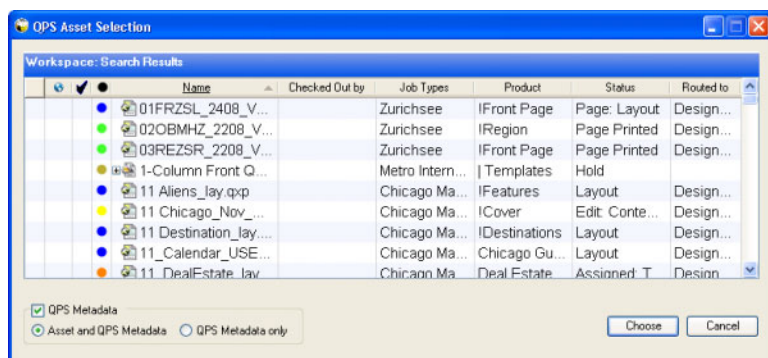
QPS Asset Selection dialog box
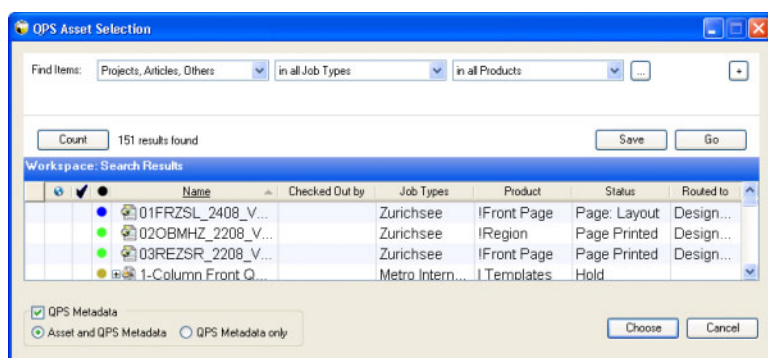
QPS Asset Selection dialog box with search controls exposed

When you select a file in QPS, the **QPS Metadata** check box becomes available. If you check this box, you can then choose whether you want to add both the file and the file's metadata to the tranformation, or add only the file's metadata to the transformation.

Select the asset you want and then click **Choose**.

7 If you choose **Document Pool**, the **QuarkXPress Server Document Pool** dialog box displays. Select the file you want and then click **OK**.
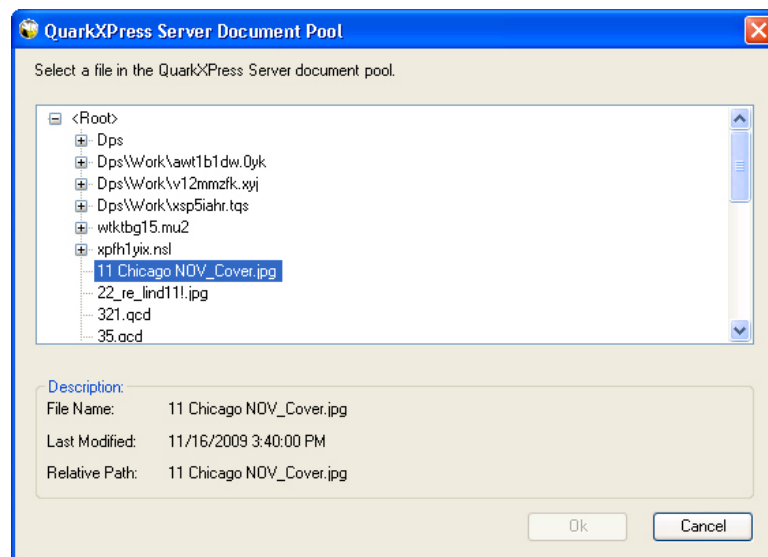
QuarkXPress Server Document Pool dialog box

8 Click **OK**. Mapping components for the schema or schemas are added to the transformation.

9 Add additional schemas if necessary, and then customize the mappings to produce the desired output. (For more information, see "*Sample transformations*.")

- To preview the the results of the transformation as XML, click the **Output** tab at the bottom of the transformation window.

- To preview the output as a rendered file from QuarkXPress Server, click **Generate Preview**. In the dialog box that displays, choose whether you want to generate the preview with a **Construct** request or a **Modify** request, then use the **Additional Parameters** field to refine the preview request using QuarkXPress Server parameters. The preview file opens in the default application for viewing files in that format.
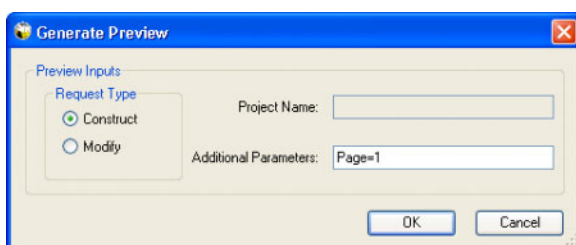
Generate Preview dialog box

➡ Choose **Construct** if you are building the entire project or article from scratch. Choose **Modify** if you are using a template.

➡ Automation Services considers each mapping component that has incoming connections but no outgoing connections to be an output.

10 To validate the current transformation, choose **File > Validate Mapping**. Automation Services checks the transformation for errors and displays the **Validation Results** dialog box. Fix any errors that display.
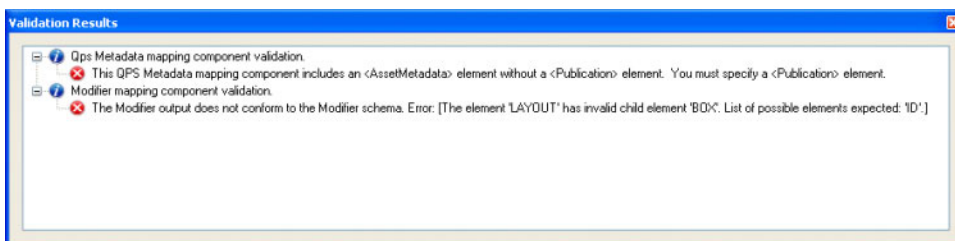
Validation Results dialog box

11 Click **Next**. The transformation is deployed to the server.

➡ You cannot use a transformation in an automation profile until you deploy it.

Mapping content to boxes and components

The topics below explain how to map content to text and picture boxes in QuarkXPress projects and how to text and picture components in QuarkCopyDesk articles.

Mapping text content

There are two ways to supply text for a text box or component:

• Map an element containing the text to the `<CONTENT>` node for the target box or component or to a `<RICHTEXT>` node.

- Map an element containing a reference to a text file to the the `<CONTENT>` node for the target box or component. The reference should follow the form `file:[URI of file]`.

  There are two ways to format mapped text:

- Map unformatted text to the `<CONTENT>` node for the target box or component. The text will use the default formatting for the target box or component.

- Map text that has been formatted using the Modifier `<RICHTEXT>` element type. For more information, see *A Guide to QuarkXPress Server* and the *QuarkXPress Server Web Integration Guide*.

  To map text content to a text box or article component:

1 Add a Modifier mapping component for the project or article that will contain the text. Select **Project** as the root element.

2 In the Modifier mapping component, set the `PROJECT/LAYOUT/BOX@BOXTYPE` attribute to "CT_TEXT".

3 In the Modifier mapping component, navigate to the `PROJECT/LAYOUT/BOX/CONTENT` element.

4 Add a mapping component for the XML file that will supply the text.

5 Map the source text element to the Modifier mapping component's `PROJECT/LAYOUT/BOX/CONTENT` element or to a `PROJECT/LAYOUT/BOX/TEXT/STORY/PARAGRAPH/RICHTEXT` element.

6 To identify which box the content should go into, supply a value for the `PROJECT/LAYOUT/BOX/ID@UID` attribute. If you are using a template and you know this value, you can specify it as a constant. If you are constructing a project from scratch, you can generate a value with an **auto-number** function.

➡ For an example of how to map text to boxes in a template using component names, see "*XML to QuarkCopyDesk with Template (sample transformation)*."

7 To verify that the content is mapping to the box correctly, click **Output** at the bottom of the mapping window.

8 To verify that the content is formatting correctly, click **Generate Preview**.

Mapping picture content

There are four ways to supply picture content for a text box or component:

- As a relative reference in an XML element.

- As an absolute reference in an XML element. If you use this method, Automation Services ignores everything except the file name and looks for the file in the location(s) specified in the automation profile.

- As binary data in the XML. Automation Services does not support this method.

- As a `<CONTENT>` element in a project or article.

To map picture content to a text box or article component:

1 Add a Modifier mapping component for the project or article that will contain the picture. Select **Project** as the root element.

2 In the Modifier mapping component, set the `PROJECT/LAYOUT/BOX@BOXTYPE` attribute to "CT_PICT". If you are mapping to an article, also set the component's `PROJECT/LAYOUT/ARTICLE/COMPONENT@COMPONENTCLASS` attribute to "CT_PICT"

3 Map the source picture element to the Modifier mapping component's `PROJECT/LAYOUT/BOX/CONTENT` element.

4 To identify which box the content should go into, supply a value for the `PROJECT/LAYOUT/BOX/ID@UID` attribute. If you are using a template and you know this value, you can specify it as a constant. If you are constructing a project from scratch, you can generate a value with an **auto-number** function.

➡ For an example of how to map text to boxes in a template using component names, see "*XML to QuarkCopyDesk with Template (sample transformation)*." The same approach can be used for pictures.

5 To verify that the content is mapping to the box correctly, click **Output** at the bottom of the mapping window.

6 To verify that the picture imports correctly, click **Generate Preview**. (Note that this will work only if the picture file is in the QuarkXPress Server document pool.)

When a deployed transformation executes:

- If the transformation checks the project or article into QPS, and the picture file is already in QPS, Automation Services creates a link between the project or article and the picture file. If the picture file is not in QPS, Automation Services checks it into the same publication and section as the project or article to which it is linked. For more information, see "*Assigning metadata to pictures*."

- If Automation Services cannot find a picture file, it inserts a picture of a red X with the same file name as the missing picture file as a visual indicator to QuarkXPress operators working on the file that the image is not yet available and needs to be created or scanned. The fact that Automation Services uses the same name as the missing file makes it easy for a layout artist to replace the X image with the correct image in QuarkXPress.

Working with QPS metadata

A QPSEntityModel mapping component represents all of the metadata that can be stored in QPS for an asset. If you want to create a transformation that checks its output into QPS, you must supply a QPSEntityModel mapping component for that output so that QPS knows what to put in the asset's metadata fields.

You can also use QPS metadata when you're creating a transformation that uses QPS assets as inputs. For example, assume you're creating a transformation that takes content from an article from QPS, flows that content into a QuarkXPress template, and then checks the resulting QuarkXPress project into QPS. You can use some or all of the article's metadata when you check in the project by mapping the article's QPSEntityModel mapping component to the project's QPSEntityModel mapping component.

Adding metadata to a transformation

To add a QPSEntityModel instance to the output of a transformation, choose **Insert > XML Schema/File**, choose **QPS Metadata Schema** from the **Choose** menu to the right of the **Schema File** field, and then click **OK**. When the **Select root element** dialog box displays, specify an asset type:

- To create metadata for a project, choose **ProjectInfo** as the root element.

- To create metadata for an article, choose **ArticleInfo** as the root element.

- To create metadata for a picture file, choose **PictureInfo** as the root element.

- To create metadata for any other type of asset, choose **AssetInfo** as the root element.
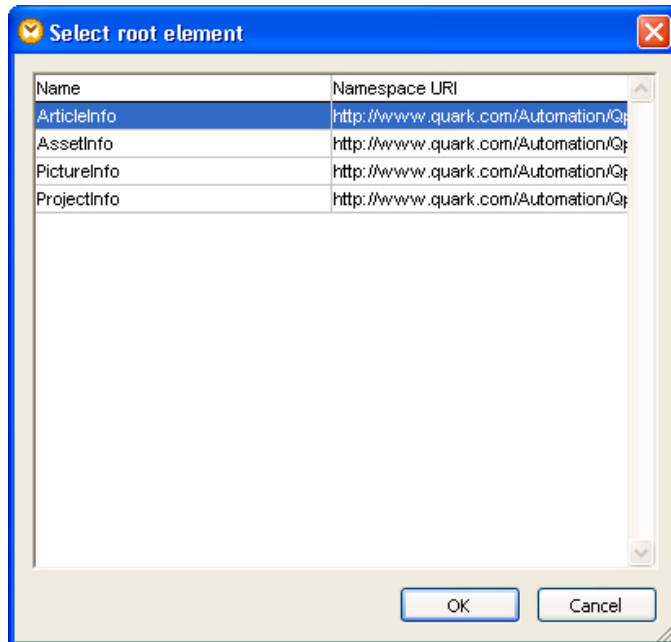
Select root element dialog box

You can also use QPSEntityModel mapping components when you're creating a transformation that uses an asset from QPS as an input. To do so, check **QPS Metadata** in the **QPS Asset Selection** dialog box when you're specifying an input file from QPS. You can add both the file and the file's metadata to the tranformation, or add only the file's metadata to the transformation.
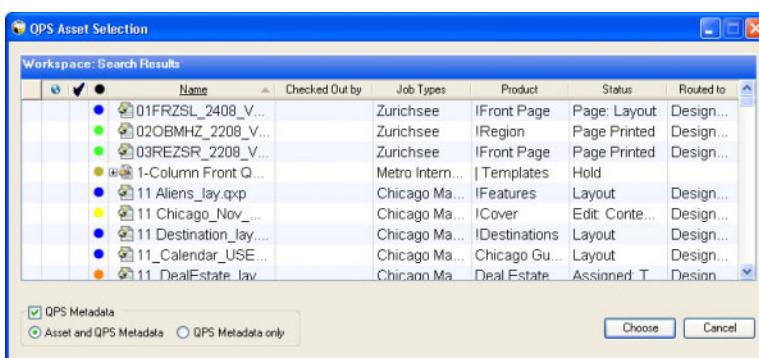
QPS Asset Selection dialog box with **Metadata** box checked

If you want to create a new revision of an existing project or article, you can simply map all of the fields in the source file's QPSEntityModel mapping component to the corresponding fields in the output file's QPSEntityModel mapping component.

Some other ways to use QPSEntityModel mapping components:

• If you're transforming a QuarkCopyDesk article into an HTML file for uploading to Drupal, you could use component names in the article's QPS metadata to control how different parts of the content (such as headlines and body text) are formatted in the HTML.

• If you're transforming an XML file to a QuarkXPress template, you can store content from the XML (such as attribute values that do not actually go into the template) in the resulting file's QPS metadata.

Assigning metadata to components

If you create article components in a transformation, you must specify metadata for each component in the article's QPSEntityModel mapping component. Each component's metadata must have a name, but all other attributes are optional. To specify a name for a component, map a value to its `ArticleInfo/Components/ComponentInfo/ComponentMetadata/Name` element.

By default, a QPSEntityModel mapping component has one component, represented by the `<ComponentInfo>` element. When you create a QPSEntityModel mapping component for an article, you can create components for that article by adding additional `<ComponentInfo>` elements. To create a new `<ComponentInfo>` element, right-click an existing `<ComponentInfo>` element and choose **Duplicate Input**.
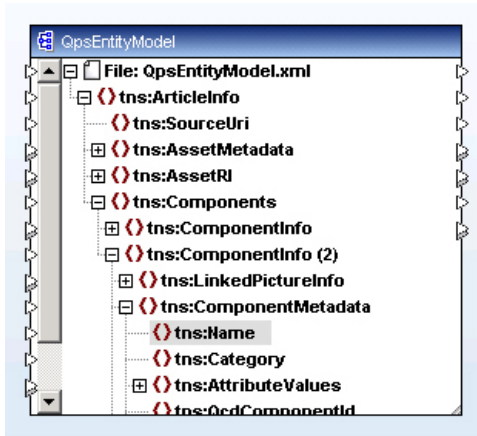
QPSEntityModel instance with two components. The `<name>` element is selected in the second component.

Assigning metadata to pictures

There are two ways to check a picture into QPS with a transformation:

- Check the picture file directly into QPS. To do so, create a `PictureInfo` QPSEntityModel mapping component and map the URI of the picture file from the source XML file to the QPSEntityModel mapping component's `<SourceUri>` element. Use the other elements in the QPSEntityModel mapping component to specify the picture's QPS metadata.

- Map the picture content to a box in a project or to a picture component in an article, then check the project or article into QPS. (For more information on mapping pictures to boxes and components, see "*Mapping picture content.*") Any pictures imported by the transformation are automatically checked into QPS.

When you check a picture in using the second method, you can specify the metadata for the picture in the QPSEntityModel mapping component for the project.
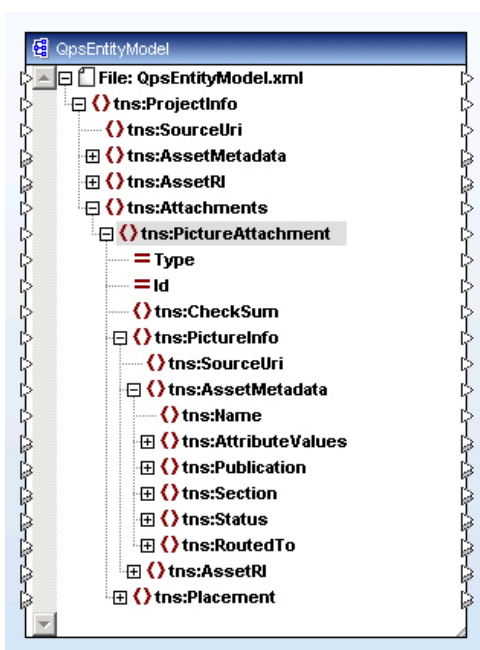
QPSEntityModel with `<PictureAttachment>` element selected

QPS metadata validation

When you use a QPSEntityModel mapping component to check an asset in to QPS, there are some element values you must provide values for. There are also some element values that will cause Automation Services to generate a warning if they are left blank. The topics below provide more detail.

➡ To validate the current transformation at any time, choose **File > Validate Mapping**.

Constraints for ProjectInfo

When you add a mapping component with a root element of `ProjectInfo` to a transformation, the following constraints apply.

Element type	Constraint
ProjectInfo/AssetMetadata/Name	Required.
ProjectInfo/AssetMetadata/Publication	Required.
ProjectInfo/AssetMetadata/Section	Required.
ProjectInfo/AssetMetadata/Status	Required.
ProjectInfo/AssetMetadata/RoutedTo	Warning generated if blank.
ProjectInfo/AssetMetadata/AttributeValues	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `ProjectInfo/AssetMetadata/Name`, and then also

Element type	Constraint
	specify a second `Name` value using `ProjectInfo/AssetMetadata/AttributeValues`, the second name is ignored.
`ProjectInfo/Components/ComponentInfo/ComponentMetadata/Name`	Required for each component.
`ProjectInfo/Components/ComponentInfo/ComponentMetadata/Type`	Required for each component.
`ProjectInfo/Components/ComponentInfo/ComponentMetadata/AttributeValues`	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `ProjectInfo/Components/ComponentInfo/ComponentMetadata/Name`, and then also specify a second `Name` value using `ProjectInfo/Components/ComponentInfo/ComponentMetadata/Name`, the second name is ignored.

Constraints for ArticleInfo

When you add a mapping component with a root element of `ArticleInfo` to a transformation, the following constraints apply.

Element type	Constraint
`ArticleInfo/AssetMetadata/Name`	Required.
`ArticleInfo/AssetMetadata/Publication`	Required.
`ArticleInfo/AssetMetadata/Section`	Required.
`ArticleInfo/AssetMetadata/Status`	Required.
`ArticleInfo/AssetMetadata/RoutedTo`	Warning generated if blank.
`ArticleInfo/AssetMetadata/AttributeValues`	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `ArticleInfo/AssetMetadata/Name`, and then also specify a second `Name` value using `ArticleInfo/AssetMetadata/AttributeValues`, the second name is ignored.
`ArticleInfo/Components/ComponentInfo/ComponentMetadata/Name`	Required for each component.
`ArticleInfo/Components/ComponentInfo/ComponentMetadata/Type`	Required for each component.
`ArticleInfo/Components/ComponentInfo/ComponentMetadata/AttributeValues`	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `ArticleInfo/Components/ComponentInfo/`

Element type	Constraint
	`ComponentMetadata/Name`, and then also specify a second `Name` value using `ArticleInfo/Components/ComponentInfo/ComponentMetadata/Name`, the second name is ignored.

Constraints for AssetInfo

When you add a mapping component with a root element of `AssetInfo` to a transformation, the following constraints apply.

Element type	Constraint
`AssetInfo/AssetMetadata/Name`	Required.
`AssetInfo/AssetMetadata/Publication`	Required.
`AssetInfo/AssetMetadata/Section`	Required.
`AssetInfo/AssetMetadata/Status`	Required.
`AssetInfo/AssetMetadata/RoutedTo`	Warning generated if blank.
`AssetInfo/AssetMetadata/AttributeValues`	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `AssetInfo/AssetMetadata/Name`, and then also specify a second `Name` value using `AssetInfo/AssetMetadata/AttributeValues`, the second name is ignored.

Constraints for PictureInfo

When you add a mapping component with a root element of `PictureInfo` to a transformation, the following constraints apply.

Element type	Constraint
`PictureInfo/AssetMetadata/Name`	Required.
`PictureInfo/AssetMetadata/Publication`	Required.
`PictureInfo/AssetMetadata/Section`	Required.
`PictureInfo/AssetMetadata/Status`	Required.
`PictureInfo/AssetMetadata/RoutedTo`	Warning generated if blank.
`PictureInfo/AssetMetadata/AttributeValues`	If you specify a QPS metadata attribute both directly and with the `AttributeValues` element type, the values specified with the `AttributeValues` element will be ignored. For example, if you specify a name with `PictureInfo/AssetMetadata/Name`, and then also specify a second `Name` value using `PictureInfo/AssetMetadata/AttributeValues`, the second name is ignored.

Understanding attachment handling

When you use Automation Services to create a new project from a template that has attachments, the attachments are removed.

When you use Automation Services to create a new revision of a project that has attachments, the attachments are preserved unless you replace the content in their boxes with new content.

Exporting transformations

You can export tranformations as packages, which you can then manually deploy on multiple servers. To export a transformation:

1 Display the **Transformation** pane.

2 Click the transformation you want to export and then choose **File > Edit Transformation**.

3 Choose **File > Export Transformation**.

4 Navigate to the desired location and then click **Export**.

For information about deploying exported transformations, see "*Deploying profiles, transformations, and workflows.*"

Working with automation profiles

The topics below describe how to create and use automation profiles.

It is important that the QPS Server you log on in QPS Connect Client is the same server as the one specified in the "AutomationServices.Server.config" file (located at `[application folder]\Server\AutomationServices_Data\Resources\Configuration`). If it is not, the assets and searches you use to build an automation profile may not be available when that automation profile executes.

Creating an automation profile

To create an automation profile:

1 In QPS Connect Client, display the **Automation** pane.
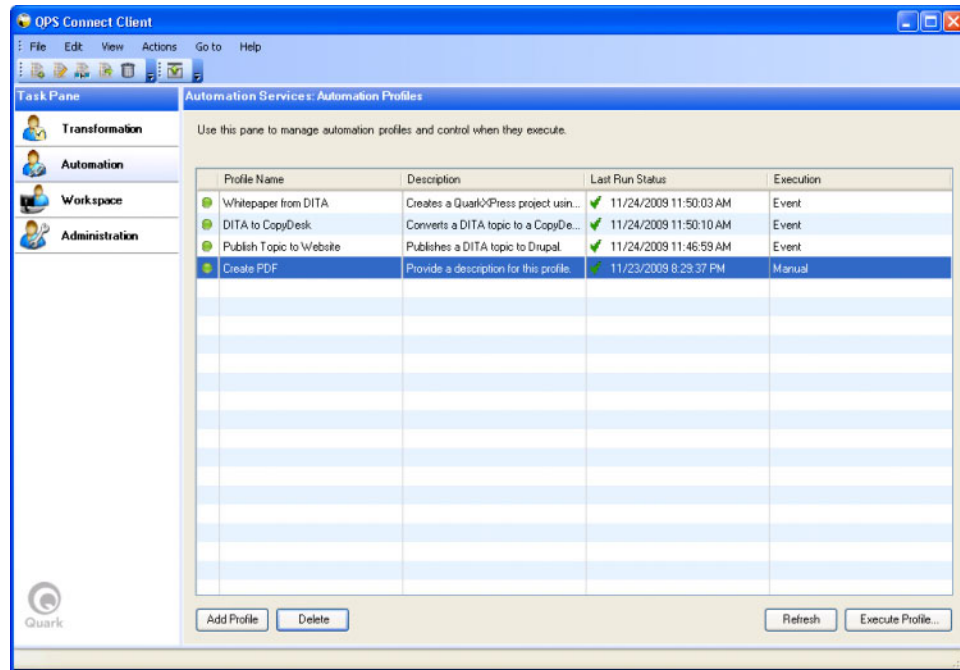
Automation pane of **QPS Connect Client** window

2 Click **Add Profile**. The **Profile** screen displays.
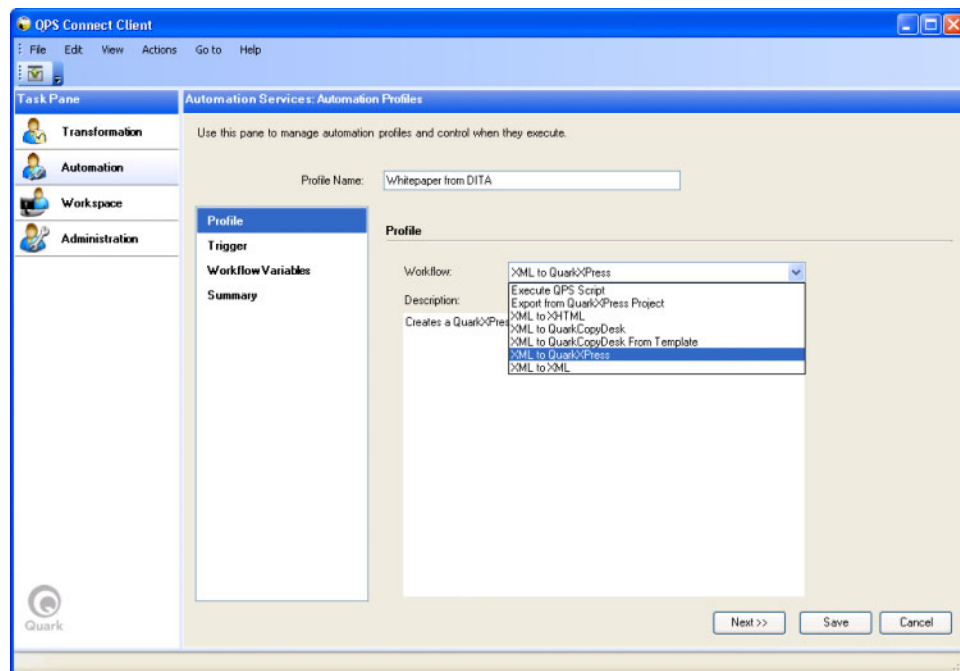
Add Profile screen of **Automation** pane

3 Choose a workflow from the **Workflow** drop-down menu, enter a name in the **Profile Name** field, enter a description in the **Description** field, and then click **Next**. The **Trigger** screen displays.
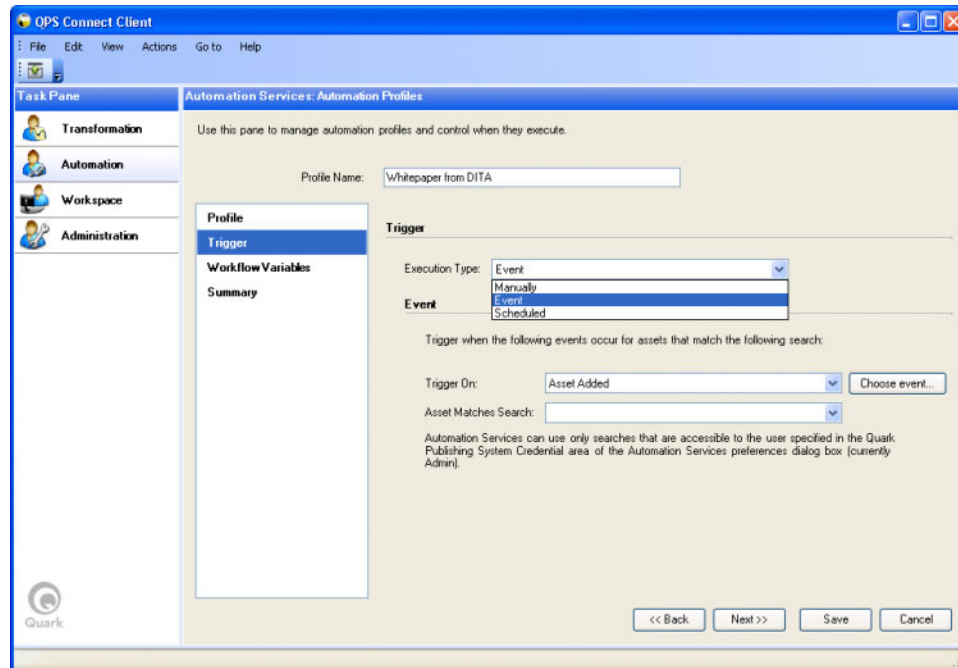
Add Scheduling screen of **Automation** pane

4   Choose an option from the **Execution Type** drop-down menu and then configure the scheduling for the automation task. The options in this screen differ depending on which option you choose.

• To trigger the automation profile at specific intervals, choose **Scheduled** from the **Execution Type** drop-down menu and then use the scheduling controls to indicate when the automation profile should execute. You can trigger an automation profile to execute once, every *x* minutes, hourly, daily, weekly, or monthly. The **Input File** field at the bottom of the **Scheduled** area lets you specify a trigger input.

WORKING WITH AUTOMATION PROFILES

Scheduled controls

- To trigger an automation profile when a QPS event occurs, choose **Event** from the **Execution Type** drop-down menu and then use the **Event** controls to control when the profile should execute. Event-driven automation profiles execute when a particular event occurs for a particular type of object that matches a particular QPS search. Use the **Object Type** drop-down menu to choose the type of object that can trigger the automation profile. Use the **Choose event** button to display a dialog box where you can indicate which events should trigger the automation profile. Choose the QPS search that indicates which assets can trigger the automation profile from the **Asset Matches Search** drop-down menu. Choose **Manage Searches** from the **Asset Matches Search** drop-down menu to display a dialog box where you can work with QPS searches.

Event controls

- To create a manually executed automation profile, choose **Manual** from the **Execution Type** drop-down menu. You must specify the trigger input when you execute the profile.

44 | A GUIDE TO AUTOMATION SERVICES 8.1

For more information, see "*Trigger screen*."

5 Click **Next**. The **Workflow Variables** screen displays.
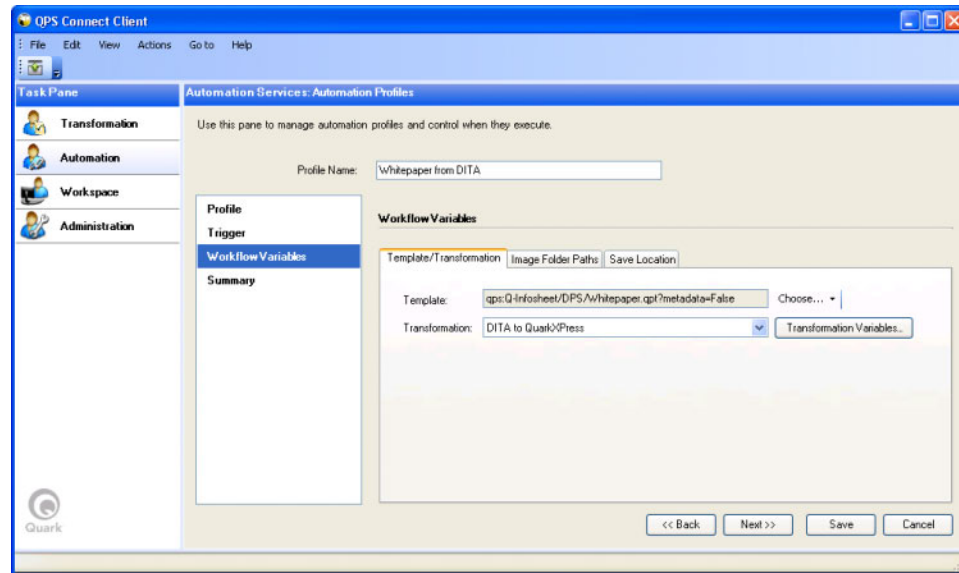
Add Parameters screen of **Automation** pane

6 Continue setting up the automation profile in the **Workflow Variables** screen. The controls displayed depend on which workflow is selected (for more information, see Automation Services *Workflow Reference*). However, the following controls are common to many workflows.

Template controls display in any workflow that uses a template.
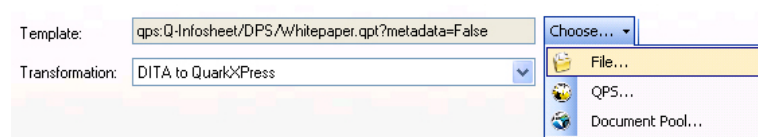
Transformation controls

Use the controls in the **Template** area to specify which template is used with the transformation.

Transformation controls display in any workflow that uses a transformation.

Transformation controls

After you choose a transformation, you must configure the variables that are used by that transformation. To do so, click **Configure Transformation Variables**. The **Configure Transformation Variables** dialog box displays.
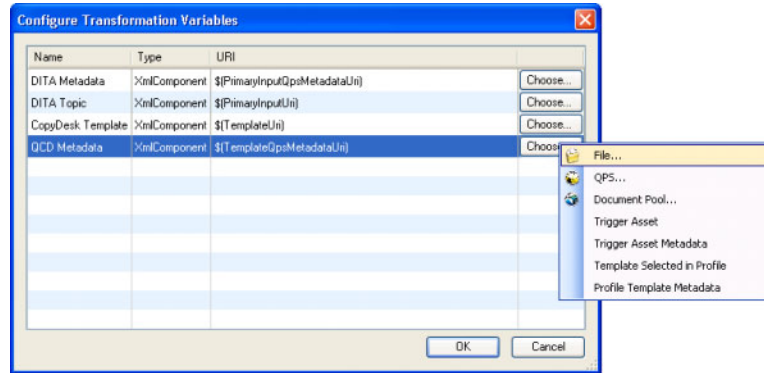
Configure Transformation Variables dialog box

This dialog box lists all of the variable inputs for the transformation. For each variable input, choose an option from the **Choose** button menu. In addition to the standard **From File System** and **From QPS Search** options, you can choose from a number of options that are specific to the workflow used by the profile. For example, the options provided by the "XML to CopyDesk Template" are:

• **Trigger Asset**: This is the asset that triggered the automation profile. In other words, this is an asset that matches the criteria in the **Event** area of the **Scheduling** dialog box.

• **Trigger Asset Metadata**: This the QPS metadata for the asset that triggered the automation profile.

• **Template Selected in Profile**: This is the template specified in the **Workflow Variables** pane.

• **Profile Template Metadata**: This is the QPS metadata for the template specified in the **Workflow Variables** pane.

➡ Custom workflows created by a System Integrator or developer may have additional custom parameters.

Image Search Folder controls display in any workflow that involves pictures.

Image Search Folder controls

Automation Services searches the folders in the **Search Paths** list for any images that are used by the files involved in the automation profile, in the order they are listed, from top to bottom. To add a folder to the list, use the **Choose** button to add its path to the **Folder Path** field, and then click the plus button to add the path to the **Search Paths** list. You can use the arrows beneath the **Search Paths** list to move the selected path up or down in the list.

➡ If you add a file system directory to the list, Automation Services will search that directory and all of its subdirectories. When searching directories in the file system, Automation Services uses the first matching picture file it finds.

➡ If you add a series of QPS publication/sections to the list, Automation Services will search them in the order you list them, from top to bottom.

➡ If Automation Services cannot find a picture file, it inserts a picture of a red X with the same file name as the missing picture file. This makes it easy for a layout artist to replace the X image with the correct image later on.

Save Location controls display in any workflow that involves saving a file.

Save Location controls

You can choose to save the file in QPS, in Drupal, or in the file system.

7 Click **Next**. The **Summary** screen displays.
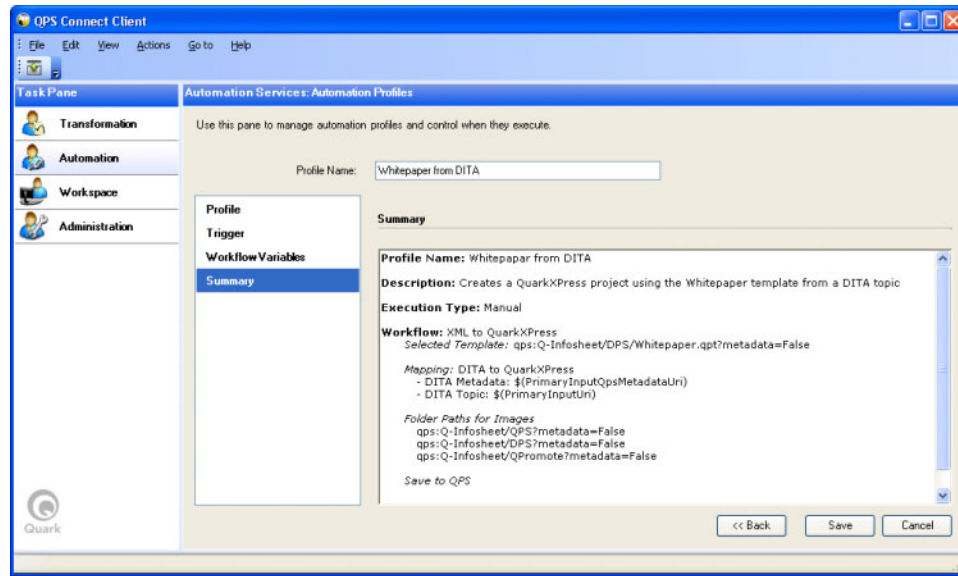
Summary screen of **Automation** pane

8 Click **Save**.

➡ You cannot use an automation profile until you set log on credentials for Automation Services. For more information, see "*Setting log on credentials*."

➡ Creating an automation profile is fairly bandwidth intensive and consequently may not work well over a slow Internet connection.

Manually executing an automation profile

If an automation profile is configured to be executed manually, you can execute it as follows.

1 In the QPS Connect Client **Automation** pane, select an automation profile that is configured to execute manually and click **Execute Profile**. The **Execute Profile** dialog box displays so that you can specify the file on which the profile should execute.
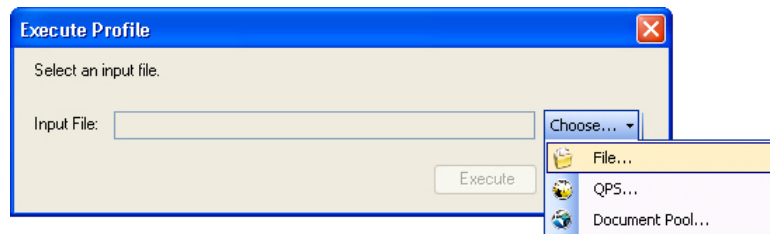
Execute Profile dialog box

2 Click the **Choose** button and choose an option from the submenu.

- If you choose **From File System**, an **Open** dialog box displays. Choose the target file in the file system.

- If you choose **From QPS Search**, a QPS search dialog box displays. Use the controls in this dialog box to find the target file.

3 If you choose a file in the file system, the **Create and Use Copy in QPS** checkbox becomes available. If you check this box, Automation Services checks a copy of the file into QPS and uses that copy.

4 Click **Execute**.

Enabling and disabling automation profiles

There may be circumstances where you need to temporarily disable an automation profile. To do so, right-click on the green circle at the beginning of the automation profile's row in the QPS Connect Client **Automation** pane and then choose **Disable** from the context menu. The circle's color changes to red, indicating that the automation profile is disabled.

You can enable disabled automation profiles the same way.

Sample transformations

The topics below describe how to create some sample transformations.

➡ For information on how to create transformations in general, see "*Working with transformations*."

XML to XHTML (sample transformation)

To create a sample XML to HTML transformation:

1 Create a new transformation.

2 Add a mapping component for an XML schema file to the transformation. Include an input XML instance for use in previewing the results of the transformation. For this example, we will use an XML file that contains text and picture information in a form similar to that of a QuarkCopyDesk article. (You could also use an actual QuarkCopyDesk article as the input.)

3 Add a mapping component for the XHTML schema to the transformation and move it to the right of the XML mapping component.

4 Map a value to the `<title>` element of the XHTML mapping component. This can be a value from the input, a constant, or any other value you want to supply.

5 To create text elements in the output, map the element type of a repeating text element to (for example) from the source mapping component to the `<p>` element type in the XHTML mapping component. A new `<p>` element will be created for each occurrence of the source element.
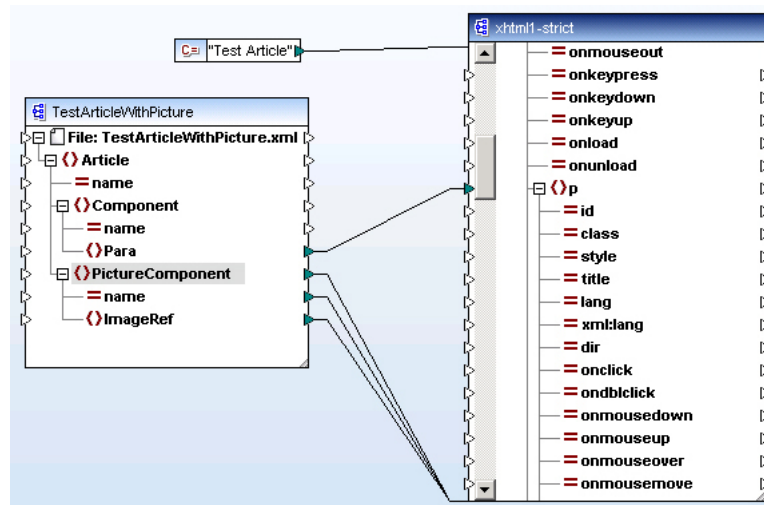
Mapping of paragraph in QuarkCopyDesk text component to `<p>` element in XHTML output

6 To create picture elements in the output, map an element type that contains picture paths to the `src` attribute of the `<img>` element type.

Mapping of picture in QuarkCopyDesk picture component to `<img>` element in XHTML output

7 To validate the transformation, choose **File > Validate Mapping**.

8 To test the output, click **Output** at the bottom of the mapping window.

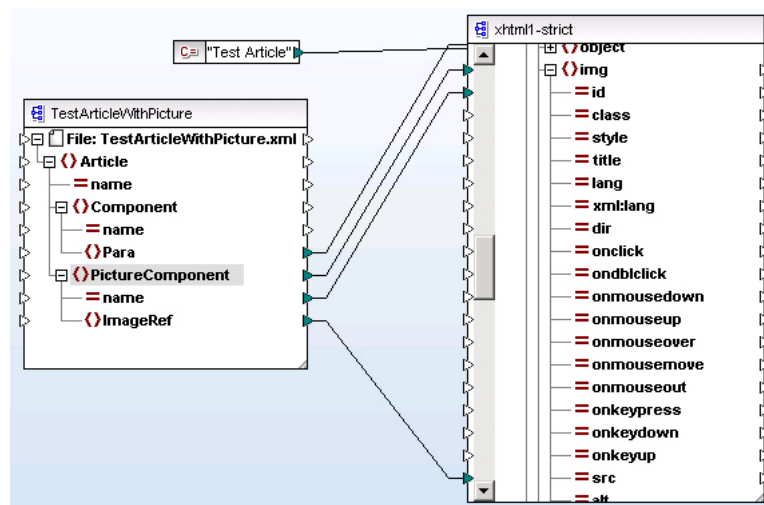## XML to QuarkCopyDesk (sample transformation)

To create a sample XML to QuarkCopyDesk transformation:

1 Create a new transformation.

2 Add a mapping component for an XML schema file to the transformation. Include an input XML instance for use in previewing the results of the transformation. For this example, we will use an XML file that contains text and picture information in a form similar to that of a QuarkCopyDesk article. (You could also use an actual QuarkCopyDesk article as the input.)

3 Add a mapping component for the Modifier schema to the transformation and move it to the right of the XML mapping component. This schema represents the output article.

4 When creating a QuarkXPress project or QuarkCopyDesk article, you must specify a unique layout ID or name. In the output mapping component, expand the `<LAYOUT>` element, then expand the `<ID>` element and map a unique value to the `NAME` or `UID` attribute.

5 When creating a QuarkCopyDesk article, you must specify a unique article ID or name. The article name is used as the file name at output. In the output mapping component, expand the `<ARTICLE>` element, then expand the `<ID>` element and map a unique value to the `NAME` or `UID` attribute.
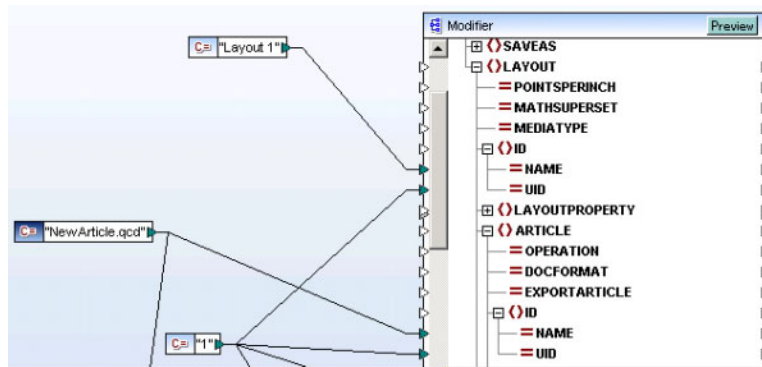
Mapping of unique values to `<ID>` elements for `<LAYOUT>` and `<ARTICLE>`

6 To create a text component in the output, expand the `<COMPONENT>` element type and map unique values to the `NAME`, `UID`, `BOXNAME`, and `BOXUID` attributes. (Note that these values must be unique only across attribute types; in the example below, the "1" constant is reused several times.) To identify the component as a text component, map the value "CT_TEXT" to the `COMPONENTCLASS` attribute.
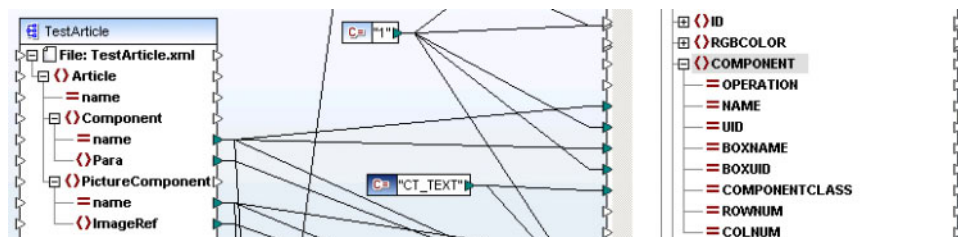
Mapping of values to text component

7 Before you can insert text into the text component, you must first supply a unique box name and ID. In the output mapping component, locate the

`PROJECT/LAYOUT/SPREAD/BOX/ID` element and map unique values to its `NAME` and `UID` attributes.
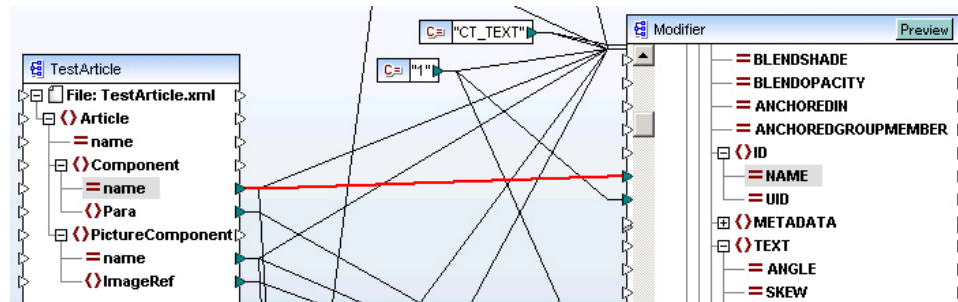
Mapping of values to box `<ID>` element

8 To insert formatted text into the text component, map a text element type to the `PROJECT/LAYOUT/SPREAD/BOX/TEXT/STORY/PARAGRAPH/RICHTEXT` element type.
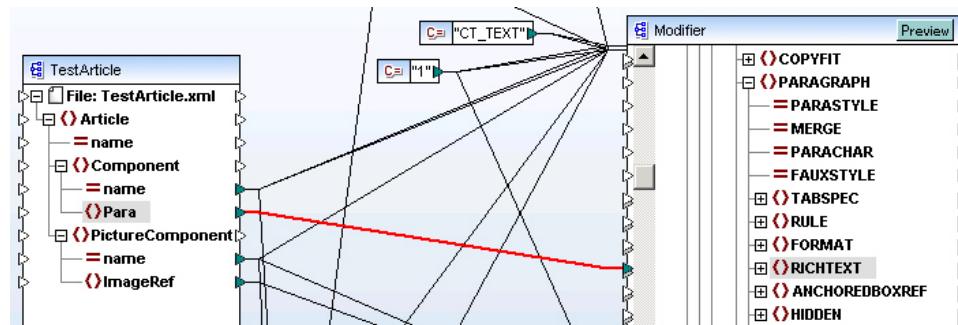
Mapping of text content to `<RICHTEXT>` element

➡ If you want to use the formatting in a template or the default paragraph style sheet in the Job Jackets file for the publication and section where you're checking in the article, you can also simply map a text element to the `PROJECT/LAYOUT/SPREAD/BOX/CONTENT` element type.

9 To create a picture component in the output, first create a new `<COMPONENT>` element. Right-click the existing `<COMPONENT>` element type and choose **Duplicate Input**. Then expand the `<COMPONENT>` element type and map unique values to the `NAME`, `UID`, `BOXNAME`, and `BOXUID` attributes. To identify the component as a picture component, map the value "CT_PICT" to the `COMPONENTCLASS` attribute.
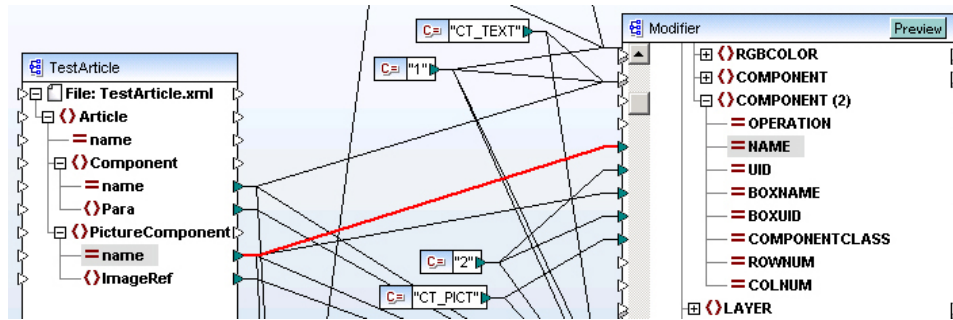
Mapping of picture in QuarkCopyDesk picture component to `<img>` element in XHTML output

10 To insert a picture into the picture component, map a picture element type to the `PROJECT/LAYOUT/SPREAD/BOX/CONTENT` element type.

Mapping of picture content to `<CONTENT>` element

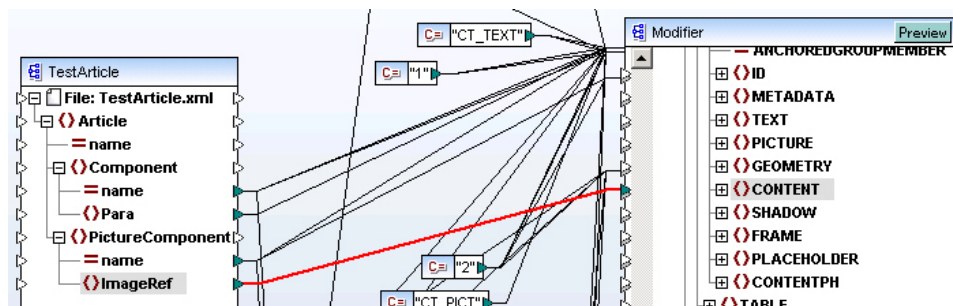**11** To validate the transformation, choose **File > Validate Mapping**.

12 To test the output, click **Output** at the bottom of the mapping window or click **Generate Preview**.

For information about creating metadata for a QuarkCopyDesk article, see "*Article metadata (sample transformation)*."

Article metadata (sample transformation)

This example picks up where "*XML to QuarkCopyDesk (sample transformation)*" leaves off. To create QPS metadata for a sample XML to QuarkCopyDesk transformation:

1 Add a QPSEntityModel mapping component to the transformation.

2 To specify the article-level metadata to be used when the article is checked in, map values to the QPSEntityModel's element types as shown below. Here, we are using the QPSEntityModel of the source article as an input to the QPSEntityModel of the output article. Note that the value for the `ArticleInfo/AssetMetadata/Name` element must match the value for the article's `PROJECT/LAYOUT/ARTICLE/ID@NAME` attribute.
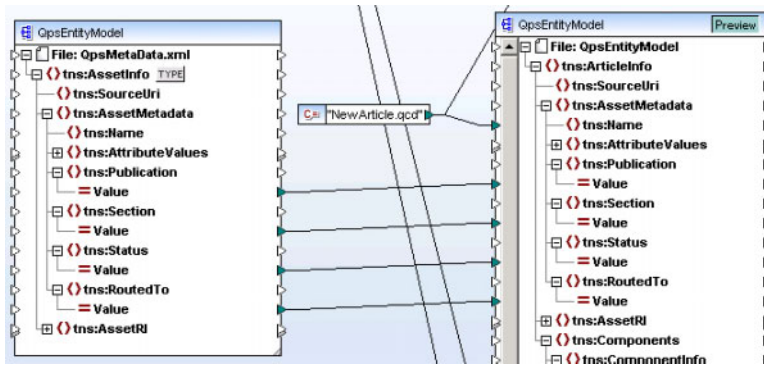
Mapping values to article metadata

3 To specify the component-level metadata to be used when the article is checked in, create a `<ComponentInfo>` element for each component in the article and map values to the components' child elements as shown below. Here, we are using the source article as the source of these values. Note that you must specify a `<Name>` and `<Category>` for each component.
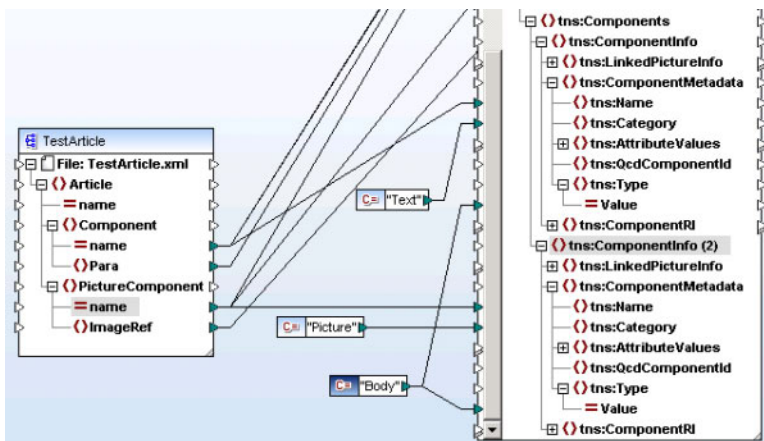
Mapping values to component metadata

4 To test the output, run the transformation.

XML to QuarkCopyDesk with Template (sample transformation)

This transformation takes content from an XML source (an XML representation of a QuarkCopyDesk file) and inserts into a new QuarkCopyDesk article based on a QuarkCopyDesk article template.

To create a sample XML to QuarkCopyDesk with Template transformation:

**1** Follow steps 1-7 in "*XML to QuarkCopyDesk (sample transformation)*."

2 Add a Modifier mapping component for the template.

3 Because you want the output to look like the template, you should start by mapping everything in the template Modifier mapping component to the output Modifier mapping component. To do so, first choose **Connection > Settings for Connect Matching Children**. When the **Settings for Connect Matching Children** dialog box displays, configure it as shown below. This lets you connect all of the elements in the two mapping components in one step.
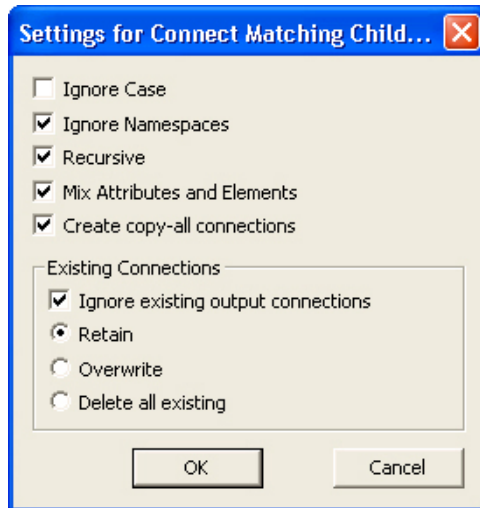
Settings for Connect Matching Children dialog box

4 Attach the template's `<PROJECT>` element type to the output article's `<PROJECT>` element type. All child element types should connect automatically.
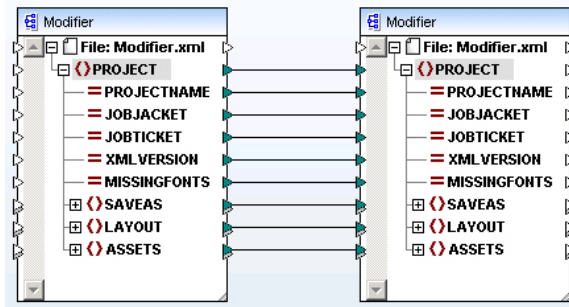
Template mapping component connected to output article mapping component

Obviously you don't want the template's content to go into the output article's components; you want the content from your input source to go into the output. To accomplish this, you will need to break some of the connections between the template and the output article and replace them with connections from the input (in this case, a different QuarkCopyDesk article). Specifically, you'll probably want to link the input mapping component's content elements to `<CONTENT>` and `<RICHTEXT>` elements in the output article.

5 To get the right text into the right box, you can use logic like that shown below. This construction says, if a component in the template is named "Body 1" and the component's BOXUID attribute is equal to a box's BOX/ID@UID attribute, then go ahead and put the content into the component's <RICHTEXT> element. In other words: Find the template component named "Body 1," find its corresponding box, and then put the text in that box. (Automation Services knows which box to use in the output because the component's NAME and BOXUID attributes are also mapped to the corresponding attributes in the output article.) You can map picture content to a picture component the same way.
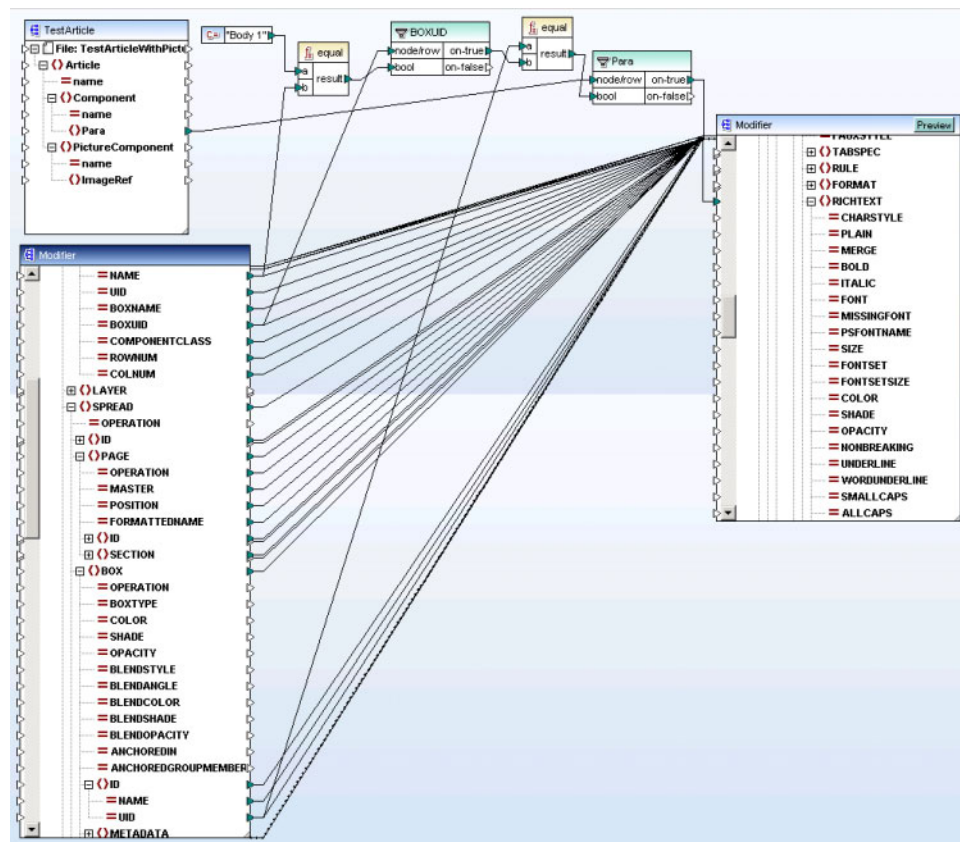
Mapping of text content to <RICHTEXT> element

6 To validate the transformation, choose **File > Validate Mapping**.

7 To test the output, click **Output** at the bottom of the mapping window or click **Generate Preview**.

Transformation best practices

This topic provides information that will help you to build successful transformations.

Mapping

When you connect a source element to a target element that has child elements, you can automatically connect any matching child elements too. To automatically map child elements, go to the **Connection** menu and make sure **AutoConnect Matching Children** is checked before you create the mapping.

If you choose to do this, keep in mind that it will make the transformation more complex and potentially slower. It's best to create only the connections you actually need.

If you automatically connect child elements for a recursive schema (for example, a schema where `<a>` can contain `<b>` and `<b>` can contain `<a>`), MapForce will continue creating recursive connections until it runs out of memory. To prevent this from happening, choose **Connection > Settings for Connect Matching Children** and uncheck **Recursive** before you make the mapping.

User-defined functions

Do not use an XML element as an input to a user-defined function in a transformation. If you do, it may initially appear to work, but the link between the element in the user-defined function and the source file will be broken when you save or deploy the transformation or reload the schemas. Also, you cannot specifyan XML element that resides in a user-defined function as a variable input.

General

Automation Services does not lock transformations while they are being edited. If you need to work on a particular transformation, make sure no one else is working on it at the same time or you may overwrite each others' changes.

You cannot use existing XSL files in a transformation. Also, you cannot import maps that have been created in a free-standing edition of MapForce. However, if you want to trigger an XSLT transformation of an asset, you can use the "Execute QPS Script" workflow to trigger a script which executes an XSL stylesheet created in standalone Altova Mapforce or any other XLS creation tool.

Changing schemas and instances

In the transformation editing window, Automation Services does not support changing the schema file or instance file by right-clicking and choosing **Properties** from the context menu. To change a schema file or instance file, click **Transformation Name and Variables**, select the input, and click **Choose**.

Setting mapping component properties

When you add a mapping component to a transformation, right-click the component, choose **Properties**, and then uncheck the following options:

- Add schema/DTD reference

- Enable input processing optimizations based on min/maxOccurs

System administration

The topics below provide information relevant to system administrators

Setting access privileges and idle time-out

To work properly, Automation Services must have sufficient access privileges to all of the save locations that you have specified in your automation profiles. It must also have a time-out value high enough to keep it running when there are no incoming requests.

To set up access privileges and idle time-out:

1 Choose or create a user account to be used by Automation Services.

2 In the IIS management console, right-click on the application pool for Automation Services, choose **Advanced Settings**, and set the identity to the Automation Services user account.
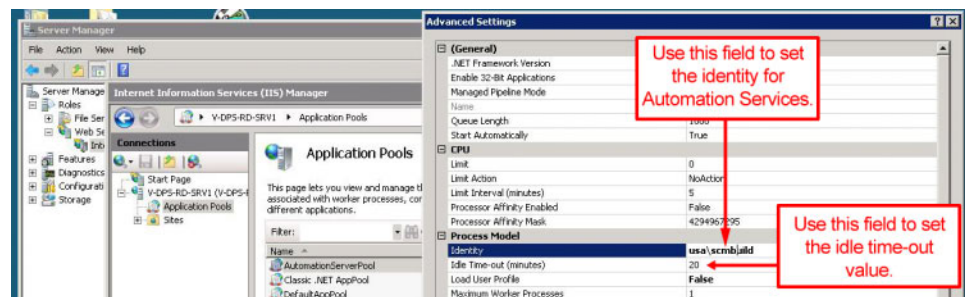
IIS management console with Automation Services application pool displayed. The **Advanced Settings** dialog box displays when you right-click the **AutomationServerPool** icon and choose **Advanced Settings**.

3 In the same dialog box, set **Idle Time-out (minutes)** to `1700`. This enables the server to keep executing automation profiles even if a long time has elapsed since the last client request.

4 Give the Automation Services user account **Co-owner** rights to the "AutomationServices_Data" folder in the Automation Services application folder. This step ensures that automation profiles can be successfully deployed.

5 Give the Automation Services user account **Co-owner** rights to all folders that automation profiles need to write to.

➡ It is *not* sufficient to assign the IIS "Network Service" user account to the Automation Services output folders and give it full control.

Configuring QuarkXPress Server

To set up QuarkXPress Server for use with Automation Services:

• Set up all instances of QuarkXPress Server to use a common document pool.

• If realm authentication is enabled, give all instances of QuarkXPress Server the same realm authentication.

• Enable the ability to return QuarkXPress projects for all instances of QuarkXPress Server (**QuarkXPress Server > Server Configuration > Server** tab > uncheck **Disable QuarkXPress Document Return**).

• Give all instances of QuarkXPress Server the permission to create folders in the document pool (**QuarkXPress Server > Server Configuration > Server** tab > check **Generate Hierarchy On Document Upload**).

Deploying profiles, transformations, and workflows

A *package* is a .zip file that contains a workflow, compiled transformation, or automation profile, along with everything it needs to be executed on the server. To deploy a package, simply put it in the `[application folder]\Server\AutomationServices_Data\Resources\Deployment` directory on the server computer. (This folder is created by the installer.) The next time you launch the server, the server installs the package in the "AutomationServices" folder in the Web server and then removes it from this folder.

After installation, the Automation Services server loads packages into a SQL Server Compact Edition database that runs in memory on the Automation Services server computer. It takes a few seconds to process deployed packages when the server starts, so users may experience a short delay while the server is starting up. Once the server instance is up and running, it automatically loads and unloads the contents of packages on the fly as it needs them.

Changing the Drupal image directory

During installation of the Automation Services server component, you can specify the location of the folder where image files generated by Automation Services should be put for Drupal. To set or change this location after installation:

1 In the "Drupal Content Uploader.php" file, set the `IMAGES_ROOT` path constant in the `Configuration` class to the directory where you want to store images generated by Automation Services (relative to the Apache root directory). Make sure this directory actually exists.

2 Open the "AutomationServices.Server.config" file (located at `[application folder]\Server\AutomationServices_Data\Resources\Configuration`), locate the "Drupal" section, and set the `imagefolderpath` entry to the same path you set `IMAGES_ROOT` to in the previous step.

Miscellaneous

This topic provides information about a variety of miscellaneous system administration topics.

XML file formats

The "AutomationServices.Server.config" file (located at `[application folder]\Server\AutomationServices_Data\Resources\Configuration`) contains an `<entry>` element with the key "xmlFileExtensions". This key is used to identify XML files, QuarkXPress project files, and QuarkCopyDesk files, respectively. If you need to support a new XML schema, you must add a new suffix to the comma-delimited list of suffixes.

IP address changes

If the IP address of the QuarkXPress Server Manager computer, the QPS Server computer, or the Drupal computer changes, you must update the appropriate section in the "AutomationServices.Server.config" file:

```
<section name="Qxpsm">
  <entry key="server">10.1.100.34</entry>
  <entry key="port">61400</entry>
  ...
</section>
  <section name="Qps">
  <entry key="server">10.1.100.34</entry>
  <entry key="port">61403</entry>
   ...
</section>
<section name="Drupal">
  <entry key="server">10.1.3.50</entry>
  <entry key="port">8080</entry>
  ...
</section>
```

➡ If Automation Services is running on the same computer as QPS Server, set the server and port for the "Qps" entry to `127.0.0.1` and `61403`.

Note that any changes you make to this file will not be honored until you recycle the Automation Services application pool in IIS.

Index

A

automation profiles 8, 13, 17, 41
AutomationServices.Server.config 62

C

construct vs. modify 31
conventions 5
credentials 16

D

deployment 61
Drupal 16

E

event-driven tasks 44
event-triggered tasks 20

F

formatting conventions 5

I

images 14, 46

L

logging on 16

M

manually executed tasks 22, 44, 48
MapForce 7, 10, 28
maps 11
metadata 33, 54
modify vs. construct 31

O

output 14, 47

P

packages 61
passwords 16
previews 25, 30

Q

QPS 16
QPS metadata 33, 54
QPSEntityModel 33
QuarkXPress Server 16

S

sample transformations 50
save locations 14, 47
scheduled tasks 20, 43
schemas 12, 27
scripting 7

T

templates 14, 55
transformation variables 26, 45
transformations 7, 10, 24, 27, 50, 58
triggers 8, 12, 13, 19, 46

V

variables, transformation 26, 45
variables, workflow 22, 45

W

workflow variables 8, 22, 45
workflows 7, 9